

SCOPE OF APPLICATION All Project/Engineering	<b>HYUNDAI</b> <b>AutoEver</b>	SHT/SHTS 1 / 52
Responsibility: Classic AUTOSAR	AUTOSAR RamTst User Manual	DOC. NO
AUTOSAR RamTst User Manual		

Document Change Histroy				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	내용(개정 전 -> 개정 후)
2016-04-29	1.0	CY Song		• Initial Creation
2016-05-24	1.1	CY Song	Chap 5.1.1	• 형식변경 • RamTstGetVersionInfoApi Defalut값 변경
2017-06-30	1.2	YJ Yun	Chap 4.3.1.1 Chap 4.3.1.3 Chap 5.1.1 Chap 9.1.2 Chap 9.2.1	• ChangeLog 추가 • Limitations 추가 • RamTstPublicNotifHeaderFile 삭제, Default value 수정 • Medium Coverage Algorithm 변경내용 수정 • Sequence 수정
2018-05-24	1.3	YJ Yun	Chap 8	• SWP Error Code 추가
2018-12-31	1.4	YJ Yun	Chap 4.3.1.1 Chap 7.2.1 9)  Chap 9	• ChangeLog 수정 • Message ERR093057 수정 • Message INF093051 삭제 • 설정 및 가이드 수정
2019-10-17	1.5	YJ Yun	Chap 5	• 설정 카테고리 수정 • ChangeLog 추가
2020-12-31	1.3.2.0	YJ Yun	Chap 4	• ChangeLog 추가
2021-01-18	1.3.3.0	YJ Yun	Chap 4	• ChangeLog 추가
2021-12-30	1.3.4.0	JH Lim	Chap 4.3 Chap 9.2.1.1	• Update change log
2022-08-19	1.3.5.0	YJ Yun	Chap 4.3 Chap 7.2.1 Chap 5.1.5	• Update change log • ERR093058, ERR093059, ERR093060 추가 • StartAddress/EndAddress 설명 보강

10th Edition Date: 19. 08. 2022	File Name RamTst_UM.pdf	Creation YJ Yun 2022/08/19	Check SI Kang 2022/08/19	Approval JH Baek 2022/08/19
Document Management System				

## Table of Contents

<b>1. OVERVIEW .....</b>	<b>5</b>
<b>2. REFERENCE.....</b>	<b>5</b>
<b>3. AUTOSAR SYSTEM.....</b>	<b>6</b>
3.1 Overview of Software Layers .....	6
3.2 AUTOSAR RamTst .....	6
<b>4. PRODUCT RELEASE NOTES .....</b>	<b>9</b>
4.1 Overview .....	9
4.2 Scope of the release .....	9
4.3 Module release notes .....	9
4.3.1 RamTst .....	9
<b>5. CONFIGURATION GUIDE .....</b>	<b>15</b>
<b>5.1 RamTst 모듈.....</b>	<b>15</b>
5.1.1 RamTstCommon Container.....	15
5.1.2 RamTstAlgorithms Container .....	17
5.1.3 RamTstConfigParams Container .....	18
5.1.4 RamTstAlgParams Container .....	18
5.1.5 RamTstBlockParams Container .....	19
5.1.6 RamTstPublishedInformation Container .....	19
5.1.7 RamTstDemEventParameterRefs Container .....	20
<b>6. APPLICATION PROGRAMMING INTERFACE (API) .....</b>	<b>20</b>
<b>6.1 Type Definitions .....</b>	<b>20</b>
6.1.1 RamTst_ExecutionStatusType.....	20
6.1.2 RamTst_TestResultType .....	20
6.1.3 RamTst_AlgParamsIdType .....	20

6.1.4	RamTst_AlgorithmType .....	21
6.1.5	RamTst_NumberOfTestedCellsType.....	21
6.1.6	RamTst_NumberOfBlocksType .....	21
<b>6.2</b>	<b>Macro Constants.....</b>	<b>21</b>
<b>6.3</b>	<b>Functions.....</b>	<b>21</b>
6.3.1	Initialization.....	21
6.3.2	De-Initialization.....	22
6.3.3	RamTst_Stop .....	22
6.3.4	RamTst_Allow .....	23
6.3.5	RamTst_Suspend .....	23
6.3.6	RamTst_Resume .....	23
6.3.7	RamTst_GetExecutionStatus .....	24
6.3.8	RamTst_GetTestResult .....	24
6.3.9	RamTst_GetTestResultPerBlock .....	25
6.3.10	RamTst_GetVersionInfo .....	25
6.3.11	RamTst_GetAlgParams .....	26
6.3.12	RamTst_GetTestAlgorithm .....	26
6.3.13	RamTst_GetNumberOfTestedCells .....	26
6.3.14	Ramtst_SelectAlgParams .....	27
6.3.15	RamTst_ChangeNumberOfTestedCells .....	27
6.3.16	RamTst_RunFullTest .....	28
6.3.17	RamTst_RunPartialTest .....	28
6.3.18	RamTst_MainFunction .....	28
<b>7.</b>	<b>GENERATOR .....</b>	<b>30</b>
<b>7.1</b>	<b>Generator Option.....</b>	<b>30</b>
<b>7.2</b>	<b>Generator Error Message .....</b>	<b>30</b>
7.2.1	Error Messages .....	30
7.2.2	Warning Messages .....	36
7.2.3	Information Messages.....	36
<b>8.</b>	<b>SWP ERROR CODE.....</b>	<b>36</b>
<b>8.1</b>	<b>SWP Error Code List .....</b>	<b>36</b>
8.1.1	RAMTST_E_RAM_FAILURE .....	36

<b>9. APPENDIX .....</b>	<b>37</b>
<b>9.1 기능별 설정 Guide.....</b>	<b>37</b>
9.1.1 Low Coverage Algorithm .....	37
9.1.2 Medium Coverage Algorithm .....	39
9.1.3 High Coverage Algorithm .....	40
<b>9.2 설계시 유의사항 .....</b>	<b>42</b>
9.2.1 Foreground Test.....	42
9.2.2 Background Test.....	45
9.2.3 RamTst_TestResultType .....	49
9.2.4 RamTst API Call-Context .....	50
9.2.5 Low Power Mode .....	50
<b>9.3 Bswmd (Bsw Module Description) .....</b>	<b>50</b>
9.3.1 MainFunction 주기 설정.....	50
9.3.2 Bsw 모듈 version 설정.....	51
<b>9.4 Exclusive Areas .....</b>	<b>51</b>
9.4.1 모듈 별 SchM Apis .....	51
9.4.2 설정방법.....	52

## 1. Overview

본 문서는 Autosar 표준 SRS / SWS 를 기반으로 작성되었으며 RamTst 모듈사용 시 보다 자세한 기능적인 설명이 필요한 경우 Reference 문서를 참조해야 한다.

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

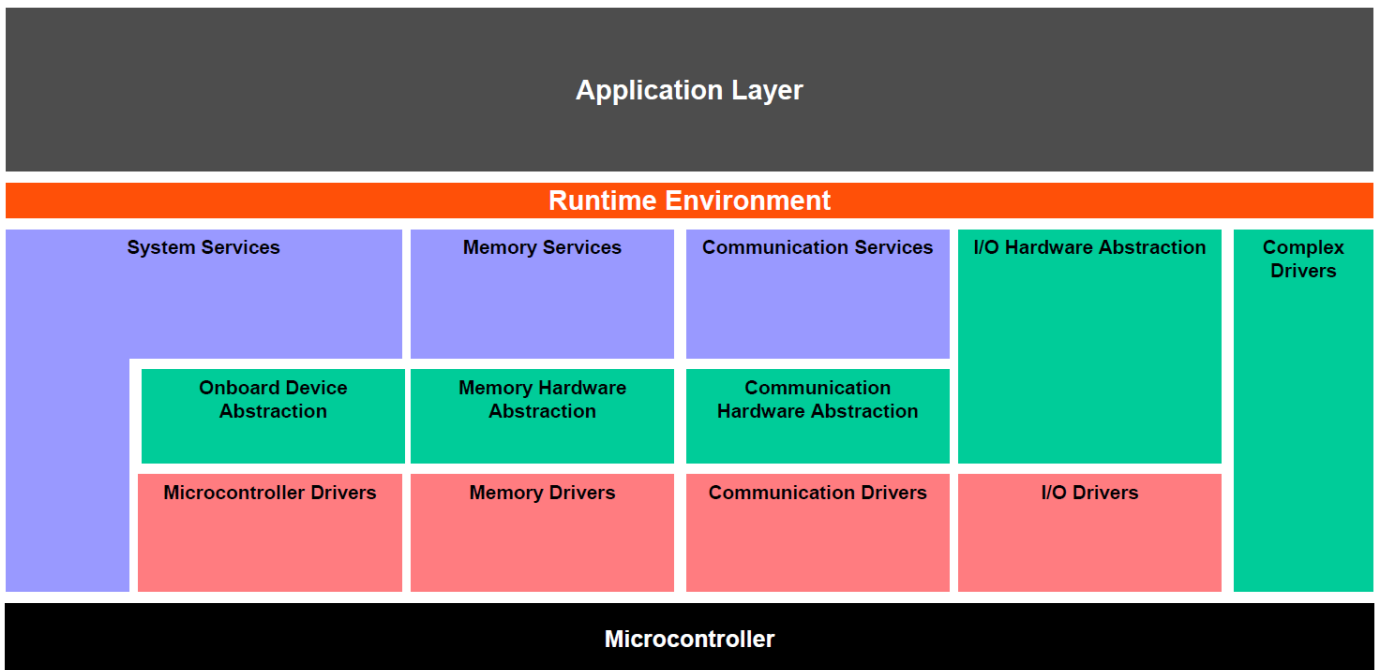
## 2. Reference

Sl. No.	Title	Version
1.	AUTOSAR_SWS_RAMTest.pdf	1.5.0 or later
2.	AUTOSAR_SRS_RAMTest.pdf	1.2.0

### 3. AUTOSAR System

#### 3.1 Overview of Software Layers

AUTOSAR 플랫폼의 Layered Architecture 는 아래와 같다. AUTOSAR 플랫폼은, Service Layer, ECU Abstraction Layer, Complex Device Drivers 및 Microcontroller Abstraction Layer 로 구분될 수 있다.



#### 3.2 AUTOSAR RamTst

The RAM Test is a test of the physical health of the RAM cells. It is not intended to test the contents of the RAM.

There are several RAM Test algorithms available. The different RAM Test algorithms are divided into 3 groups of diagnostic coverage rates:

Group 1 (Low): Diagnostic coverage rate = <60%

Group 2 (Medium): Diagnostic coverage rate = 60% - 90%

Group 3 (High): Diagnostic coverage rate = 90% - ≤99%.

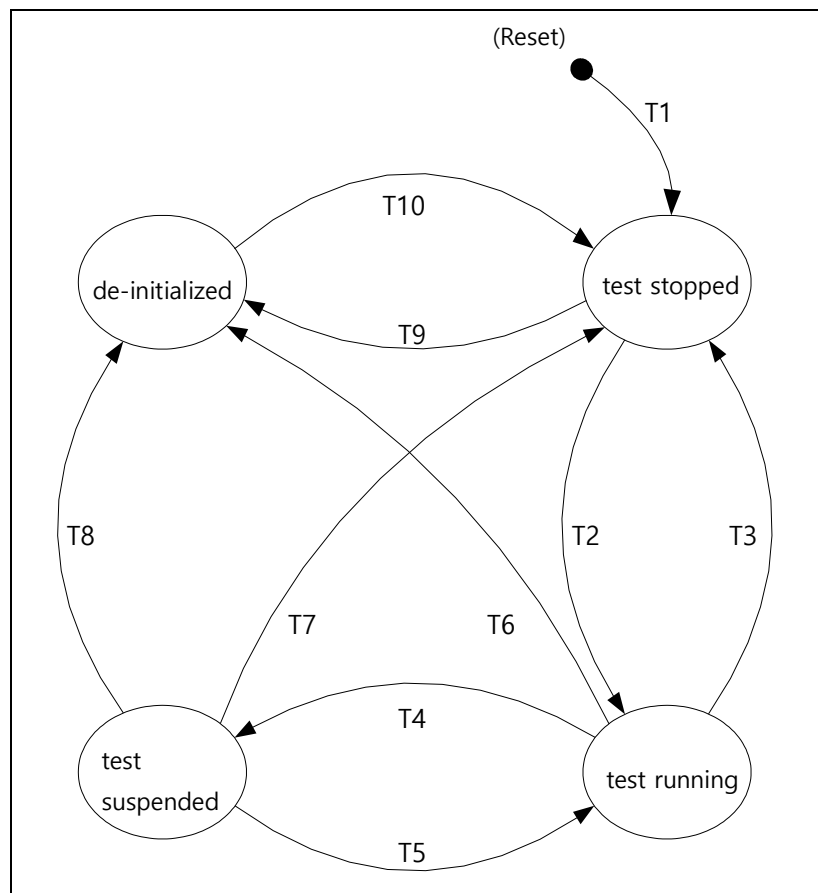
It should be noted, that these coverage rates are only rough estimates, which in the concrete case depend on the fault model for the specific hardware and on the detailed implementation of the test algorithm.

An ECU safety analysis must be performed to determine which RAM Test diagnostic coverage rate (Low, Medium or High) is required. Appropriate RAM Test algorithms and further configuration parameters are then selected at compile time. At run time, the application software may choose between the compiled algorithms (and between further parameters).

A RAM Test may be called synchronously by the test environment (hereafter called "foreground test") or may be called in a cyclic manner by an OS task or other cyclic calling method (hereafter called "background test").

For background testing, RamTst\_MainFunction() is called periodically by a scheduler, and is interruptible. One complete test consists of testing with one algorithm over the memory space defined by the currently selected configuration. This complete test is split up over many scheduled calls.

For foreground testing, RamTst\_RunFullTest() or RamTst\_RunPartialTest() is called once, and is not interruptible by routines which access the tested memory area (this has to be controlled by the test environment). It tests with one algorithm over the memory space (or a subset in case of partial test) defined by the selected configuration.



Event	Event Trigger
T1	API: RamTst_Init
T2	API: RamTst_RunFullTest API: RamTst_RunPartialTest API: RamTst_Allow
T3	API: RamTst_Stop (or end of RamTst_RunFullTest) (or end of RamTst_RunPartialTest)
T4	API: RamTst_Suspend (or end of RamTst_RunPartialTest)
T5	API: RamTst_Resume API: RamTst_RunPartialTest
T6	API: RamTst_DeInit
T7	API: RamTst_Stop
T8	API: RamTst_DeInit
T9	API: RamTst_DeInit
T10	API: RamTst_Init

Note: The state "test running" does not necessarily mean that testing is continuously being performed. For foreground testing, it does mean that the test is directly per-formed by an API call and RamTst\_MainFunction() is not scheduled. For back-ground testing, it only means that RamTst\_MainFunction() is permitted to test a small portion of the RAM when it is called periodically by the scheduler.



## 4. Product Release Notes

### 4.1 Overview

This chapter aims to provide release related contents for Hyundai AutoEver RamTst Product, RamTst Software product release version, restrictions and specifics are described.

### 4.2 Scope of the release

이 문서에 대한 모든 내용은, 다음의 RamTst 모듈에 한정한다.

Module	Autosar version	SWS version	Module version
RamTst	4.0.3	1.5.0	1.3.5.0

※ Module version 은 각 모듈의 BswModule Description(Bswmd)파일의 Sw version 을 의미한다.

### 4.3 Module release notes

#### 4.3.1 RamTst

##### 4.3.1.1 Change Log

###### ➤ Version 1.3.5.0 (2022-08-20)

###### - 신규 기능

###### ■ N/A

###### - 개선 사항

###### ■ UNECE Cyber Security 법규 대응을 위한 보안 코딩 개선

원인	UNECE Cyber Security 법규 대응을 위한 보안 코딩 개선
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

###### - 개선 사항

###### ■ Generator Validation 기능 개선

원인	Test Block 의 StartAddress,EndAddress 에 대한 Validation 기능 개선
동작 영향	(1) StartAddress

	- Low, Medium 알고리즘: Cell size(32bits)로 align 되어야 함. - High 알고리즘: CRC type(8,16,32 bits)에 맞춰 align 되어야함 (2) EndAddress -High 알고리즘 : CRC type(8,16,32 bits)에 맞춰 align 되어야함
설정 영향	없음
ASW 조치 사항	Validation error 발생시 설정 수정 필요

➤ Version 1.3.4.0 (2021-12-30)

- 신규 기능

■ N/A

- 개선 사항

■ UNECE Cyber Security 법규 대응을 위한 보안 코딩 개선

원인	UNECE Cyber Security 법규 대응을 위한 보안 코딩 개선
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

- 개선 사항

■ RamTst Init 및 ForeGround Test 구동 시 BswM Det 발생의 건 개선

원인	BswM 이 Init 되기 이전에 RamTst Init 에서 Full Test 수행하는데, LowPower 모드 천이 방지를 위한 BswM API 를 호출하여 Det 발생
동작 영향	없음
설정 영향	없음
ASW 조치 사항	* 4.3.1.2 Limitation 참조 하여 Full Test 나 FGTest 는 StartUp 구간에서만 호출될 수 있도록 한다 (LowPower 모드와의 충돌을 방지)

- 개선 사항

■ Generation 할때마다 생성 파일이 변경되는 현상 개선

원인	Generator 내에 정렬 기능 추가가 필요
동작 영향	Generation 시 생성파일의 코드 순서가 바뀌지 않는다.
설정 영향	없음
ASW 조치 사항	없음

- 오류 수정

■ MCU 의 Endian 을 고려하지 않도록 개선

원인	RamTst 의 사용자는 코드도 MCU 에서 적어주는 값이기 때문에 RamTst 는 별도로 Endian 을 고려하지 않아도 된다.
동작 영향	없음
설정 영향	없음
ASW 조치 사항	기존에 Endian 을 역으로 고려하여 TC 를 만들었던 사용자는, 역으로 고려지 않도록 수정한다. 계산해서 나온 Crc 값을 그대로 코드에 반영하도록 한다.

➤ Version 1.3.3.0 (2021-01-20)

- 신규 기능

■ N/A

- 개선 사항

■ MISRA

원인	MISRA 적용 및 정당화
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

➤ Version 1.3.2.0 (2020-12-30)

- 신규 기능

■ N/A

- 개선 사항

■ RamTst Notification 함수 NULL\_PTR 설정 가능

원인	Background Test 를 사용하지 않더라도 Notification 함수 설정이 필요했었다.
동작 영향	없음
설정 영향	Background Test 를 사용하지 않을 경우 RamTst/RamTstCommon/ RamTstConfigParams / RamTstTestCompletedNotification, RamTstTestErrorNotification 에 NULL_PTR 설정 가능
ASW 조치 사항	Background Test 를 사용하지 않는 다면 위 두 paramter 에

	NULL_PTR 을 입력해야 한다.
--	---------------------

■ SWP Error Code 추가

원인	SWP Error Code(Dem Event) RAMTST_E_RAM_FAILURE 설명 추가
동작 영향	없음
설정 영향	없음
ASW 조치 사항	Appendix capter 참조

■ 설정 및 사용자 가이드 보강

원인	사용자의 이해를 돕기 위해 설정 및 사용자 가이드 보강
동작 영향	없음
설정 영향	Appendix chapter 참조
ASW 조치 사항	Appendix chapter 참조

➤ Version 1.3.0.0 (2019-10-17)

- 신규 기능

■ N/A

- 개선 사항

■ Parameter Definition File Category 수정

원인	소스 코드 오픈
동작 영향	없음
설정 영향	유저가 설정 가능한 설정에 한하여 권한 변경
ASW 조치 사항	Category 가 Changeable 로 변경된 설정 수정 가능

➤ Version 1.3.0 (2017-06-30)

- 신규 기능

■ N/A

- 개선 사항

■ Ram Test 도중 LowPowerMode 진입 방지 로직 추가

원인	LowPowerMode 에서 지원되지 않음
동작 영향	Test 도중 LowPowerMode 진입이 안됨
설정 영향	없음
ASW 조치 사항	BackGround Test 도중 LowPower Mode 진입 필요 시 Stop API 호출 (Chap 8.2.5) 참고

- Callback 설정시 extern 선언된 파일을 Include 하지 않아도 되도록 수정

원인	설정 편리성
동작 영향	없음
설정 영향	PublicNotifHeaderFile 설정 삭제
ASW 조치 사항	없음

- Medium Coverage Test Algorithm 변경

원인	H/W 적 인접 Bit 간섭영향 고려
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

- Test Start Address 가 0 으로 설정이 되지 않도록 수정

원인	0 은 null pointer 이므로 Start Address 로 설정 시 테스트되지 않음. 따라서 설정할 수 없도록 변경함
동작 영향	없음
설정 영향	없음
ASW 조치 사항	없음

➤ Version 1.2.6 (2016-05-24)

- User Manual 수정

➤ Version 1.2.5 (2016-05-20)

- Ram 전체를 Pattern Check 할 수 있도록 수정

#### 4.3.1.2 Limitations

- Spec 에서 정의된 Test Algorithm (RAMTST\_ABRAHAM\_TEST, RAMTST\_CHECKERBOARD\_TEST, RAMTST\_GALPAT\_TEST, RAMTST\_MARCH\_TEST, RAMTST\_TRANSP\_GALPAT\_TEST, RAMTST\_WALK\_PATH\_TEST)이 지원되지 않는다.  
대신, ISO 26262 규정에 맞게 Low, Medium, High Coverage Alrorphism 을 지원한다
- RamTst\_RunFullTest, RamTst\_RunPartialTest API : foreground test 는 startup 단계에서만 수행하도록 한다.
- 테스트 수행 시 오직 RamTst 만 RAM 영역을 독점적으로 액세스해야 한다.  
Foreground test 중에는 DMA, NMI, Interrupt 등도 금지되어야 한다.
- 멀티코어의 공유메모리의 경우, interrupt locking 을 해도 메모리 셀에 독점점으로 접근하는 것이 불가능할 수 있다. 이런 경우 공유 메모리 블록에 대한 테스트는 foreground test 를 이용해야 한다.  
또한 System designer 는 독점적으로 접근 가능한 시점에 foreground test 를 수행해야 한다.

#### 4.3.1.3 Deviation

None

## 5. Configuration Guide

### 5.1 RamTst 모듈

#### 5.1.1 RamTstCommon Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RamTstAllowApi <sup>1)</sup>	User Defined	C
RamTstChangeNumOfTestedCellsApi <sup>2)</sup>	User Defined	C
RamTstDevErrorDetect	User Defined	C
RamTstGetAlgParamsApi <sup>3)</sup>	User Defined	C
RamTstGetExecutionStatusApi <sup>4)</sup>	User Defined	C
RamTstGetNumberOfTestedCellsApi <sup>5)</sup>	User Defined	C
RamTstGetTestAlgorithmApi <sup>6)</sup>	User Defined	C
RamTstGetTestResultApi <sup>7)</sup>	User Defined	C
RamTstGetTestResultPerBlockApi <sup>8)</sup>	User Defined	C
RamTstGetVersionInfoApi	User Defined	C
RamTstResumeApi <sup>9)</sup>	User Defined	C
RamTstRunFullTestApi <sup>10)</sup>	User Defined	C
RamTstRunPartialTestApi <sup>11)</sup>	User Defined	C
RamTstSelectAlgParamsApi <sup>12)</sup>	User Defined	C
RamTstStopApi <sup>13)</sup>	User Defined	C
RamTstSuspendApi <sup>14)</sup>	User Defined	C
RamTstParityBit <sup>15)</sup>	User Defined (From SRS)	C
RamTstChkTransitionOneToZeo <sup>16)</sup>	User Defined (From SRS)	C
RamTstCrcAlgorithm <sup>17)</sup>	User Defined (From SRS)	C

- 1) RamTst\_Allow 함수는 RamTst\_MainFunction 이 Testing 하도록 하는 함수이며, 호출 시에 RamTst 의 execution 상태를 RAMTST\_EXECUTION\_STOPPED 에서 RAMTST\_EXECUTION\_RUNNING 으로 변경시킨다.
- 2) RamTst\_ChangeNumberOfTestedCells 함수는 Background Test 에서 RamTst\_MainFunction 한번에 Test 되는 Cell 의 개수를 변경시키는 함수이다. RamTst 의 상태가 RAMTST\_EXECUTION\_STOPPED 가 아니면 DET Error 가 발생한다.

- 3) RamTst\_GetAlgParams 함수는 현재 저장되어 있는 Alg Parameter 의 ID 를 return 한다.
- 4) RamTst\_GetExecutionStatus 함수는 RamTst 의 execution status 를 return 한다.
- 5) RamTst\_GetNumberOfTestedCells 함수는 현재 저장된 RamTst\_MainFunction 한번에 Test 되는 Cell 의 개수를 return 한다.
- 6) RamTst\_GetTestAlgorithm 함수는 현재 저장되어 있는 Test algorithm 을 return 한다.
- 7) RamTst\_GetTestResult 함수는 현재 Ram Test 의 결과값을 return 한다.
- 8) RamTst\_GetTestResultPerBlock 함수는 요청된 Block 의 Test 결과값을 return 한다.
- 9) BackGround Test 에서 상태가 suspend 일 때 RamTst\_Resume 함수를 호출하면 다시 Test 를 시작한다. execution 상태를 RAMTST\_EXECUTION\_SUSPENDED 에서 RAMTST\_EXECUTION\_RUNNING 로 변경시킨다.
- 10) RamTst\_RunFullTest 함수는 저장된(선택된) Algorithm 에 대한 모든 Block 을 Test 한다. 호출 시에 RamTst 의 execution 상태가 RAMTST\_EXECUTION\_STOPPED 가 아니면 Det Error 를 발생시키며, 시작할 때 RAMTST\_EXECUTION\_RUNNING 으로 상태 변형 후 끝나면 RAMTST\_EXECUTION\_STOPPED 로 상태를 변경한다. 그 결과는 RamTst\_GetTestResult API 로 확인 가능하다.
- 11) RamTst\_RunPartialTest 함수는 저장된(선택된) Block ID 의 Block 을 Test 한다. 호출 시에 RamTst 의 execution 상태가 RAMTST\_EXECUTION\_STOPPED, RAMTST\_EXECUTION\_SUSPENDED 가 아니면 Det Error 를 발생시키며, 시작할 때 RAMTST\_EXECUTION\_RUNNING 으로 상태 변형 후 끝나면 이전 상태로 변경시킨다. 그 결과는 RamTst\_GetTestResultPerBlock API 로 확인 가능하다.
- 12) RamTst\_SelectAlgParams 함수는 Test 할 Block 의 Algorithm parameter 를 선택하는 함수이다. 호출 시에 RamTst 의 execution 상태가 RAMTST\_EXECUTION\_STOPPED 가 아니면 Det Error 가 발생한다. 이 함수는 test result 를 초기화 시킨다. 이 함수로 Algorithm 을 선택한 후 RunFullTest 를 실행시킬 수 있다.
- 13) RamTst\_Stop 함수가 호출되면, BackGround Test (RamTst\_MainFunction 에서의 Test)를 끝낸다. 호출 시에 RamTst 의 execution 상태가 RAMTST\_EXECUTION\_RUNNING, RAMTST\_EXECUTION\_SUSPENDED 가 아니면 Det Error 를 발생시킨다.
- 14) RamTstSuspend 함수가 호출되면, BackGround Test (RamTst\_MainFunction 에서의 Test)를 일시적으로 멈춘다. 호출 시에 RamTst 의 execution 상태가 RAMTST\_EXECUTION\_RUNNING, Det Error 를 발생시키며, 호출 후에는 상태를 RAMTST\_EXECUTION\_SUSPENDED 로 변경한다.



- 15) RamTstParityBit 설정은 Low Coverage Test 를 진행할 시에 Test 할 Parity 의 종류 (Even Parity 또는 Odd Parity)를 의미한다.
- 16) Medium Coverage Test 는 Pattern Check 로 설정된 영역(ram)을 0 으로 쓰고 0 인지를 확인 후에 1 일 쓰고 1 인지 확인한다. RamTstChkTransitionOneToZero 설정은 1 인지 확인 후에 다시 한번 0 으로 쓰고 0 인지를 확인여부를 결정하는 설정이다.
- 17) High Coverage Test 를 할 경우 CRC Check 를 하는데, RamTstCrcAlgorithm 설정을 통해 어떤 CRC Algorithm 을 사용할지를 결정하는 설정이다.

## 5.1.2 RamTstAlgorithms Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RamTstAbrahamTestSelected	-	N
RamTstCheckerboardTestSelected	-	N
RamTstGalpatTestSelected	-	N
RamTstMarchTestSelected	-	N
RamTstTranspGalpatTestSelected	-	N
RamTstWalkPathTestSelected	-	N
RamTstLowCoverageTestSelected <sup>1)</sup>	User Defined (From SRS)	C
RamTstMediumCoverageTestSelected <sup>2)</sup>	User Defined (From SRS)	C
RamTstHighCoverageTestSelected <sup>3)</sup>	User Defined (From SRS)	C

- 1) RamTstLowCoverageTestSelected 는 Ram 을 Test 할때, Parity Check 를 이용할 경우에 True 로 선택한다.
- 2) RamTstMediumCoverageTestSelected 는 Ram 을 Test 할때, Pattern Check 를 이용할 경우에 True 로 선택한다.
- 3) RamTstHighCoverageTestSelected 는 Ram 을 Test 할때, CRC 를 이용할 경우에 True 로 선택한다.

### 5.1.3 RamTstConfigParams Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RamTstDefaultAlgParamsId <sup>1)</sup>	User Defined	C
RamTstMinNumberOfTestedCells <sup>2)</sup>	User Defined	C
RamTstNumberOfAlgParamSets <sup>3)</sup>	User Defined	C
RamTstTestCompletedNotification <sup>4)</sup>	User Defined	C
RamTstTestErrorNotification <sup>5)</sup>	User Defined	C

- 1) RamTstDefaultAlgParamsId 는 RamTst 모듈 Init 시에 선택할 AlgParamsId 를 의미한다. 이 값은 RamTst\_SelectAlgParams 함수로 변경이 가능하다.
- 2) RamTstMinNumberOfTestedCells 는 BackGround Test 시에 한번의 RamTst\_MainFunction 에서 처리하는 최소 Cell 의 개수를 의미한다. (Cell 의 크기는 RamTstPublishedInformation container 에서 확인이 가능하다)
- 3) RamTstNumberOfAlgParamSets 는 RamTstAlgParams Container 의 개수를 의미한다.
- 4) RamTstTestCompletedNotification 은 BackGround Test 시에 Error 없이 완료되었을 때, RamTst 모듈에서 호출하는 함수를 적는다. Background Test 를 사용하지 않을 경우 "NULL\_PTR"을 입력할 수 있다.
- 5) RamTstTestErrorNotification 은 BackGround Test 시에 Error 발생시, RamTst 모듈에서 호출하는 함수를 적는다. Background Test 를 사용하지 않을 경우 "NULL\_PTR"을 입력할 수 있다.

### 5.1.4 RamTstAlgParams Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RamTstAlgParamsId <sup>1)</sup>	User Defined	C
RamTstAlgorithm	-	N
RamTstExtNumberOfTestedCells <sup>2)</sup>	User Defined	C
RamTstMaxNumberOfTestedCells <sup>3)</sup>	User Defined	C
RamTstNumberOfBlocks <sup>4)</sup>	User Defined	C
RamTstNumberOfTestedCells <sup>5)</sup>	User Defined	C
RamTstAlgorithmCoverage <sup>6)</sup>	User Defined	C

- 1) RamTstAlgParamsId 는 RamTstAlgParams 의 ID 로 다른 RamTstAlgParams 와 겹치지 않게 Sequential 하게 설정한다.
- 2) RamTstExtNumberOfTestedCells 는 RamTstMaxNumberOfTestedCells 값과 RamTstNumberOfTestedCells 값의 절대적인 최대값을 의미한다.

- 3) RamTstMaxNumberOfTestedCells 는 BackGround Test 시에 한번의 RamTst\_MainFunction 에서 Test 하는 최대 Cell 의 개수를 의미한다.
- 4) RamTstNumberOfBlocks 은 하위 RamTstBlockParams Container 의 개수를 의미한다.
- 5) RamTstNumberOfTestedCells 한번에 test 를 하는 Cell 의 개수의 초기값을 의미한다. 이 값은 RamTst\_ChangeNumberOfTestedCells Api 를 통하여 변경할 수 있다.
- 6) RamTstAlgorithmCoverage 는 이 AlgParam 에서 Test 하는 Test Algorithm 을 의미한다.

#### 5.1.5 RamTstBlockParams Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RamTstBlockId <sup>1)</sup>	User Defined	C
RamTstStartAddress <sup>2)</sup>	User Defined	C
RamTstEndAddress <sup>3)</sup>	User Defined	C
RamTstFillPattern <sup>4)</sup>	User Defined	C
RamTstTestPolicy <sup>5)</sup>	User Defined	C

- 1) Ramtst 는 Block 단위로 Test 를 실행한다. 이 Block 의 ID 를 의미하며, (다른 Alg\_Params 의 Block ID 를 포함하여) 다른 Block ID 와 겹치지 않게 Sequential 하게 설정한다.
- 2) RamTstStartAddress 는 Test 할 Block 의 시작주소를 의미한다. RamTstCellSize 에 따른 Align 을 맞춰주어야 한다.  
High 알고리즘을 사용하는 블록의 경우는 CRC type 에 따른 align 을 맞춰야 한다.
- 3) RamTstEndAddress 는 Test 할 Block 의 끝 주소를 의미한다.  
High 알고리즘을 사용하는 블록의 경우는 CRC type 에 따른 align 을 맞춰야 한다.
- 4) RamTstFillPattern 은 RamTstTestPolicy 가 destructive 일때, Pattern Check 이후에 Ram 에 쓰는 값을 의미한다. Non\_destructive 일 경우에는 Pattern check 하기 이전 값으로 원복한다.
- 5) RamTstTestPolicy 는 Ram Test 위하여 Ram 값을 변경하였다면, Test 가 끝난 후에 Ram 값을 Test 이전값으로 원복할지 아니면 설정해둔 값(FillPattern)으로 채울지를 결정하는 설정 값이다. RAMTEST\_NON\_DESTRUCTIVE 를 선택한다면 Test 이전 값으로 원복을 의미하고, RAMTEST\_DESTRUCTIVE 를 선택한다면, Test 가 끝난 후에 FillPattern 값으로 채운다.

**Note: RamTstTestPolicy 는 오직 medium algorithm(pattern check)에만 적용된다.**

#### 5.1.6 RamTstPublishedInformation Container

다음 설정을 참고한다.

Parameter Name	Value	Category
----------------	-------	----------

Parameter Name	Value	Category
RamTstCellSize <sup>1)</sup>	32	F

1) Test 의 기본 단위인 Cell 의 크기(bit)를 설정하는 설정 값으로 4byte 로 지원한다.

#### 5.1.7 RamTstDemEventParameterRefs Container

다음 설정을 참고한다.

Parameter Name	Value	Category
RAMTST_E_RAM_FAILURE	RAMTST_E_RAM_FAILURE	C

## 6. Application Programming Interface (API)

### 6.1 Type Definitions

#### 6.1.1 RamTst\_ExecutionStatusType

Type:	Enumeration		
Range	RAMTST_EXECUTION_UNINIT	0x00	The RAM Test is not initialized or not usable.
	RAMTST_EXECUTION_STOPPED	0x01	The RAM Test is stopped and ready to be started in foreground or to be allowed in back-ground.
	RAMTST_EXECUTION_RUNNING	0x02	The RAM Test is currently running.
	RAMTST_EXECUTION_SUSPENDED	0x03	The background RAM Test is waiting to be re-sumed.
Description:	This is a status value returned by the API service RamTst_GetExecutionStatus().		

#### 6.1.2 RamTst\_TestResultType

Type:	Enumeration		
Range	RAMTST_RESULT_NOT_TESTED	0x00	The RAM Test is not executed.
	RAMTST_RESULT_OK	0x01	The RAM Test has been tested with OK result
	RAMTST_RESULT_NOT_OK	0x02	The RAM Test has been tested with NOT-OK result.
	RAMTST_RESULT_UNDEFINED	0x03	The RAM Test is currently running.
Description:	This is a status value returned by the API service RamTst_GetTestResult().		

#### 6.1.3 RamTst\_AlgoParamsIdType

Type:	uint8
-------	-------

Range	0...255	-	-
Description:	Data type used to identify a set of configuration parameters for a test algorithm.		

#### 6.1.4 RamTst\_AlgorithmType

Type:	Enumeration		
Range	RAMTST_ALGORITHM_UNDEFINED	0x00	Undefined algorithm (uninitialized value)
	RAMTST_ALGORITHM_COVERAGE_LOW	0x01	Low Coverage algorithm
	RAMTST_ALGORITHM_COVERAGE_MEDIUM	0x02	Medium Coverage algorithm
	RAMTST_ALGORITHM_COVERAGE_HIGH	0x03	High Coverage algorithm
Description:	This is a value returned by the API service RamTst_GetTestAlgorithm().		

※ RamTstAlgorithms Container 에서 Select 된 Algorithm 만 Define 이 생성된다.

예를 들어 Medium Coverage 만 선택했다면, 다음과 같이 Type 이 생성된다.

typedef enum

{

RAMTST\_ALGORITHM\_UNDEFINED,

RAMTST\_ALGORITHM\_COVERAGE\_MEDIUM,

}RamTst\_AlgorithmType;

#### 6.1.5 RamTst\_NumberOfTestedCellsType

Type:	uint32		
Range	1...(2^32-1)	-	-
Description:	Data type of number of tested RAM cells		

#### 6.1.6 RamTst\_NumberOfBlocksType

Type:	uint16		
Range	1...65535	-	-
Description:	Data type used to identify or count RAM blocks given in the test configuration parameters.		

## 6.2 Macro Constants

None

## 6.3 Functions

### 6.3.1 Initialization

<b>Function Name</b>	RamTst_Init
<b>Syntax</b>	FUNC(void, RAMTST_CODE)RamTst_Init(void)
<b>Service ID</b>	0x00
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service for RAM Test initialization.
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 6.3.2 De-Initialization

<b>Function Name</b>	RamTst_DelInit
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_DelInit(void)
<b>Service ID</b>	0x0C
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	RamTst_DelInit is used for RAM Test deinitialization.
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 6.3.3 RamTst\_Stop

<b>Function Name</b>	RamTst_Stop
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_Stop(void)
<b>Service ID</b>	0x02
<b>Sync/Async</b>	Asynchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None

<b>Return Value</b>	None
<b>Description</b>	Service for stopping the RAM Test.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.4 RamTst-Allow

<b>Function Name</b>	RamTst-Allow
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst-Allow(void)
<b>Service ID</b>	0x03
<b>Sync/Async</b>	Asynchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	This service will start the ram test if it is previously stopped by API RamTst_Stop
<b>Preconditions</b>	The Ram test module must be initialized
<b>Configuration Dependency</b>	None

## 6.3.5 RamTst-Suspend

<b>Function Name</b>	RamTst-Suspend
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst-Suspend(void)
<b>Service ID</b>	0x0d
<b>Sync/Async</b>	Asynchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service for suspending current operation of background RAM Test, until RESUME is called.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.6 RamTst-Resume

<b>Function Name</b>	RamTst-Resume
----------------------	---------------

<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_Resume(void)
<b>Service ID</b>	0x0e
<b>Sync/Async</b>	Asynchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service for allowing to continue the background RAM Test at the point is was suspended.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.7 RamTst\_GetExecutionStatus

<b>Function Name</b>	RamTst_GetExecutionStatus
<b>Syntax</b>	FUNC(RamTst_ExecutionStatusType, RAMTST_CODE) RamTst_GetExecutionStatus(void)
<b>Service ID</b>	0x04
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_ExecutionStatusType
<b>Description</b>	Service returns the current RAM Test execution status.
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 6.3.8 RamTst\_GetTestResult

<b>Function Name</b>	RamTst_GetTestResult
<b>Syntax</b>	FUNC(RamTst_TestResultType, RAMTST_CODE) RamTst_GetTestResult(void)
<b>Service ID</b>	0x05
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None



<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_TestResultType
<b>Description</b>	Service returns the current RAM Test result.
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 6.3.9 RamTst\_GetTestResultPerBlock

<b>Function Name</b>	RamTst_GetTestResultPerBlock
<b>Syntax</b>	FUNC(RamTst_TestResultType, RAMTST_CODE) RamTst_GetTestResultPerBlock(RamTst_NumberOfBlocksType BlockID)
<b>Service ID</b>	0x06
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Parameters (In)</b>	RamTst_NumberOfBlocksType
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_TestResultType
<b>Description</b>	Service returns the current RAM Test result for the specified block.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.10 RamTst\_GetVersionInfo

<b>Function Name</b>	RamTst_GetVersionInfo
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_GetVersionInfo (P2VAR(Std_VersionInfoType, AUTOMATIC, RAMTST_APPL_DATA)versioninfo)
<b>Service ID</b>	0x0A
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	versioninfo
<b>Return Value</b>	None
<b>Description</b>	This Service returns the version information of Ram Test Module.
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 6.3.11 RamTst\_GetAlgParams

<b>Function Name</b>	RamTst_GetAlgParams
<b>Syntax</b>	FUNC(RamTst_AlgorithmType, RAMTST_CODE) RamTst_GetAlgParams(void)
<b>Service ID</b>	0x12
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_AlgorithmType
<b>Description</b>	Service returns the ID of the current RAM Test algorithm parameter set.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.12 RamTst\_GetTestAlgorithm

<b>Function Name</b>	RamTst_GetTestAlgorithm
<b>Syntax</b>	FUNC(RamTst_AlgorithmType, RAMTST_CODE) RamTst_GetTestAlgorithm(void)
<b>Service ID</b>	0x07
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_AlgorithmType
<b>Description</b>	Service returns the current RAM Test algorithm.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.13 RamTst\_GetNumberOfTestedCells

<b>Function Name</b>	RamTst_GetNumberOfTestedCells
<b>Syntax</b>	FUNC(RamTst_NumberOfTestedCellsType, RAMTST_CODE) RamTst_GetNumberOfTestedCells(void)
<b>Service ID</b>	0x09
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant

<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	RamTst_NumberOfTestedCellsType
<b>Description</b>	Service returns the current number of tested cells per main-function cycle.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.14 Ramtst\_SelectAlgParams

<b>Function Name</b>	RamTst_SelectAlgParams
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_SelectAlgParams(RamTst_AlgorithmIdType newAlgParamsId)
<b>Service ID</b>	0x0B
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	RamTst_AlgorithmIdType
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service used to set the test algorithm and its parameter set.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.15 RamTst\_ChangeNumberOfTestedCells

<b>Function Name</b>	RamTst_ChangeNumberOfTestedCells
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_ChangeNumberOfTestedCells(RamTst_NumberOfTestedCellsType NewNumberOfTestedCells)
<b>Service ID</b>	0x08
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	RamTst_NumberOfTestedCellsType
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service changes the current number of tested cells.

<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.16 RamTst\_RunFullTest

<b>Function Name</b>	RamTst_RunFullTest
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_RunFullTest(void)
<b>Service ID</b>	0x10
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service for executing the full RAM Test in the foreground.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.17 RamTst\_RunPartialTest

<b>Function Name</b>	RamTst_RunPartialTest
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_RunPartialTest(RamTst_NumberOfBlocksType BlockId)
<b>Service ID</b>	0x11
<b>Sync/Async</b>	Synchronous
<b>Reentrancy</b>	Non Reentrant
<b>Parameters (In)</b>	RamTst_NumberOfBlocksType
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	Service for testing one RAM block in the foreground.
<b>Preconditions</b>	The Ram test module must be initialized.
<b>Configuration Dependency</b>	None

## 6.3.18 RamTst\_MainFunction

<b>Function Name</b>	RamTst_MainFunction
<b>Syntax</b>	FUNC(void, RAMTST_CODE) RamTst_MainFunction(void)
<b>Service ID</b>	0x01
<b>Sync/Async</b>	None
<b>Reentrancy</b>	None
<b>Parameters (In)</b>	None
<b>Parameters (Inout)</b>	None
<b>Parameters (Out)</b>	None
<b>Return Value</b>	None
<b>Description</b>	This is Scheduled function for executing the RAM Test in the background
<b>Preconditions</b>	None
<b>Configuration Dependency</b>	None

## 7. Generator

### 7.1 Generator Option

None

### 7.2 Generator Error Message

#### 7.2.1 Error Messages

- 1) ERR093001  
: Unexpected Error Found. Please contact AUTRON AUTOSAR Support System. This is an Unexpected Error. On the occurrence of this error contact AUTRON AUTOSAR Support System.
- 2) ERR093002  
: Unexpected Error Found. This error may be due to the incorrect configuration of the element(s) 'Element Name'. If the error is not resolved, then please contact AUTRON AUTOSAR Support System. This error occurs, if the structure fields that are to be generated in the C Source file are empty. Contact AUTRON AUTOSAR Support System.
- 3) ERR093003  
: 'RamTst' Component is not present in the input file(s). This error occurs, if RAMTST component is not present in any of the input ECU Configuration Description File(s).
- 4) ERR093005  
: The parameter 'Parameter Name' in the container 'Container Name' should be configured. This error occurs, if any of the mandatory configuration parameters mentioned below is not configured in ECU Configuration Description File.

Container Name	Parameter Name
BSW-IMPLEMENTATION	AR-RELEASE-VERSION
	VENDOR-ID
	SW-VERSION
BSW-MODULE-DESCRIPTION	MODULE-ID
RamTstCommon	RamTstAllowApi
	RamTstChangeNumOfTestedCellsApi

Container Name	Parameter Name
	RamTstDevErrorDetect
	RamTstGetAlgParamsApi
	RamTstGetExecutionStatusApi
	RamTstGetNumberOfTestedCellsApi
	RamTstGetTestAlgorithmApi
	RamTstGetTestResultApi
	RamTstGetTestResultPerBlockApi
	RamTstGetVersionInfoApi
	RamTstResumeApi
	RamTstRunFullTestApi
	RamTstRunPartialTestApi
	RamTstSelectAlgParamsApi
	RamTstStopApi
	RamTstSuspendApi
RamTstAlgorithms	RamTstLowCoverageTestSelected
	RamTstMediumCoverageTestSelected
	RamTstHighCoverageTestSelected
RamTstConfigParams	RamTstDefaultAlgParamsId
	RamTstMinNumberOfTestedCells
	RamTstNumberOfAlgParamSets
	RamTstTestCompletedNotification

Container Name	Parameter Name
	RamTstTestErrorNotification
RamTstAlgParams	RamTstAlgParamsId
	RamTstAlgorithmCoverage
	RamTstExtNumberOfTestedCells
	RamTstMaxNumberOfTestedCells
	RamTstNumberOfBlocks
	RamTstNumberOfTestedCells
RamTstBlockParams	RamTstBlockId
	RamTstEndAddress
	RamTstFillPattern
	RamTstStartAddress
	RamTstTestPolicy
RamTstPublishedInformation	RamTstCellSize

5) ERR093006

: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should follow the pattern 'Pattern'. This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' does not follow the pattern 'Pattern'.

6) ERR093013

: The reference path <Reference Path> provided for the parameter 'RAMTST\_E\_RAM\_FAILURE' in the container 'RamTstDemEventParameterRefs' having short name <ShortName> is incorrect.  
This error occurs, if reference path provided for the parameter RAMTST\_E\_RAM\_FAILURE in the container RamTstDemEventParameterRefs is incorrect.

7) ERR093051

: Multiplicity of the container 'RamTstBlockParams' should be equal to the value of the parameter 'RamTstNumberOfBlocks' in the container 'RamTstAlgParams'.  
This error occurs, if multiplicity of the container RamTstBlockParams is not equal to the value of the



parameter RamTstNumberOfBlocks in the container RamTstAlgParams.

8) ERR093052

: Value configured for the parameter 'Parameter Name' should be unique in the container 'Container Name'.

This error occurs, if value configured for the parameter 'Parameter Name' is not unique in the container 'Container Name'.

9) ERR093057

: If the background test is used, the parameters 'RamTstTestCompletedNotification' and 'RamTstTestErrorNotification' in the container 'RamTstCommon/ RamTstConfigParams' should be configured other than <NULL\_PTR>.

Parameter Name	Container Name
RamTstBlockId	RamTstBlockParams
RamTstAlgParamsId	RamTstAlgParams

10) ERR093053

: The value of the parameter 'Parameter Name' in the container 'Container Name' should start with <1>.

This error occurs, if the value of the parameter 'Parameter Name' in the container 'Container Name' does not start with <1>.

Parameter Name	Container Name
RamTstBlockId	RamTstBlockParams
RamTstAlgParamsId	RamTstAlgParams

11) ERR093054

: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should be sequential.

This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' is not sequential.

Parameter Name	Container Name
RamTstBlockId	RamTstBlockParams
RamTstAlgParamsId	RamTstAlgParams

12) ERR093055

: Value of the parameter 'RamTstCellSize' in the container 'RamTstPublishedInformation' should be 8, 16 or 32.

This error occurs, if value of the parameter 'RamTstCellSize' in the container 'RamTstPublishedInformation' is 8, 16 nor 32.

13) ERR093056

: Parameter 'RamTstAlgorithmCoverage' in the container 'RamTstAlgParams' should be configured as 'Configuration Name', When the value of the parameter 'Parameter Name' in the container 'RamTstAlgorithms' is configured as <true/1>.

This error occurs, if Parameter RamTstAlgorithmCoverage in the container RamTstAlgParams is not configured as 'Configuration Name' when the value of the parameter 'Parameter Name' in the container RamTstAlgorithms is configured as <true/1>.

Configuration Name	Parameter Name
RAMTST_ALGORITHM_COVERAGE_HIGH	RamTstHighCoverageTestSelected
RAMTST_ALGORITHM_COVERAGE_MEDIUM	RamTstMediumCoverageTestSelected
RAMTST_ALGORITHM_COVERAGE_LOW	RamTstLowCoverageTestSelected

14) ERR093017

: Value of the parameter 'RamTstEndAddress' in the container 'RamTstBlockParams' should be greater than the value of the parameter 'RamTstStartAddress' in the container 'RamTstBlockParams'.

This error occurs, if value of the parameter RamTstEndAddress in the container RamTstBlockParams is less than the value of the parameter RamTstStartAddress in the container RamTstBlockParams.

15) ERR093021

: Multiplicity of the container 'RamTstAlgParams' should be equal to the value of the parameter 'RamTstNumberOfAlgParamSets' in the container 'RamTstConfigParams'.

This error occurs, if multiplicity of the container RamTstAlgParams is not equal to the value of the parameter RamTstNumberOfAlgParamSets in the container RamTstConfigParams.

16) ERR093023

: Parameter 'Dependent Parameter Name' in the container 'Dependent container Name' should be configured, since value of the parameter 'Parameter Name' in the container 'Container Name' is configured as <true/1>.

This error occurs, if Parameter 'Dependent Parameter Name' in the container 'Dependent container Name' is not configured when value of the parameter 'Parameter Name' in the container 'Container Name' is configured as <true/1>.

Dependent Parameter Name	Dependent Container Name	Parameter Name	Container Name
RamTstParityBit	RamTstCommon	RamTstLowCoverageTestSelected	RamTstAlgorithms
RamTstChkTransitionOneToZero		RamTstMediumCoverageTestSelected	
RamTstCrcAlgorithm		RamTstHighCoverageTestSelected	

17) ERR093058

: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should be aligned with the Cell Size.

This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' is not aligned with Cell size (ex 32 bits).

Parameter Name	Container Name
RamTstStartAddress	RamTstBlockParams

18) ERR093059

: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should be aligned with the Crc Size.

This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' is not aligned with CRC size (ex 8/16/32 bits)..

Parameter Name	Container Name
RamTstStartAddress	RamTstBlockParams

19) ERR093060

: The value configured for the parameter 'Parameter Name' in the container 'Container Name' should be aligned with the Crc Size.

This error occurs, if the value configured for the parameter 'Parameter Name' in the container 'Container Name' is not aligned with CRC size (ex 8/16/32 bits)..

Parameter Name	Container Name
RamTstEndAddress	RamTstBlockParams

## 7.2.2 Warning Messages

None

## 7.2.3 Information Messages

### 1) INF093015

: AUTOSAR Release version <value> configured for the parameter 'AR-RELEASE-VERSION' in provided MDT file is not correct. AUTOSAR Release version should be one of the following: 4.0.3.

This information occurs, if the value of the element AR-RELEASE-VERSION present in the Bsw Module Description template is configured other than 4.0.3.

### 2) INF093001

: Value of the parameter 'Parameter Name' in the container 'Container Name' is ignored, since the value of the parameter 'Dependent Parameter Name' in the container 'Dependent Container Name' is configured as <false/0>.

This information occurs, if value of the parameter 'Parameter Name' in the container 'Container Name' is configured when the value of the parameter 'Dependent Parameter Name' in the container 'Dependent Container Name' is configured as <false/0>.

Dependent Parameter Name	Dependent Container Name	Parameter Name	Container Name
RamTstLowCoverageTestSelected	RamTstAlgorithms	RamTstParityBit	RamTstCommon
RamTstMediumCoverageTestSelected		RamTstChkTransitionOn eToZeo	
RamTstHighCoverageTestSelected		RamTstCrcAlgorithm	

## 8. SWP Error Code

### 8.1 SWP Error Code List

#### 8.1.1 RAMTST\_E\_RAM\_FAILURE

ErrorId Symbol	RAMTST_E_RAM_FAILURE
Description	Ram Test 실패시 발생
문제 발생 원인	H/W

Platform default Action	NO RESET
기능적 영향	Ram 이 물리적으로 파손되어 의도하지 않은 동작 발생
타 모듈 연관성	없음
MCU	공통
문제 유형	설정, 코드
Application 적용 가능 대책	RAM 에 H/W 적인 오류가 발생한 상태이므로 RAM H/W 검토가 필요하다.

## 9. Appendix

### 9.1 기능별 설정 Guide

※ 아래 설정들은 사용자의 이해를 돕기 위한 샘플이므로 참고용으로만 사용해야 한다.

#### 9.1.1 Low Coverage Algorithm

Path: ... > RamTstCommon [RamTstCommon] > RamTstConfigParams [RamTstConfigParams] > AlgParams\_LOW [RamTstAlgParams]

Container Details - RamTstAlgParams

Short Name*	AlgParams_LOW
Id*	1
Algorithm Coverage*	RAMTST_ALGORITHM_COVERAGE_LOW (1)
Ext Number Of Tested Cells*	5
Max Number Of Tested Cells*	4
Number Of Blocks*	1 (2)
Number Of Tested Cells*	1

Low Coverage Algorithm(1)은 설정된 Block(2)들을 Cell 단위로 Parity Check 를 수행한다.

Container Details - RamTstBlockParams

Short Name\*: RamTstBlockParams0

Block Id\*: 1

③ Start Address\*: 0xFEBC9E7C

④ End Address\*: 0xFEBC9E7F

Fill Pattern\*: 0

Test Policy\*: RAMTEST\_NON\_DESTRUCTIVE

각 Block 의 Size 는 RamTstBlockParams Container 의 Start Address(③), End Address(④)에 설정한다.

Path: RamTst [RamTst] > RamTstCommon [RamTstCommon]

Navigator

- RamTst
  - RamTstCommon
    - RamTstAlgorithms
    - RamTstConfigParams
      - Alg Params [3]
        - AlgParams\_LOW
          - Block Params [1]
            - RamTstBlockParams0
        - AlgParams\_MED
          - Block Params [2]
        - AlgParams\_HIGH
          - Block Params [1]
      - RamTstPublishedInformation
      - RamTstDemEventParameterRefs

Get Test Result Api\*: ☒ true

Get Test Result Per Block Api\*: ☒ true

Get Version Info Api\*: ☐ false

Resume Api\*: ☐ false

Run Full Test Api\*: ☒ true

Run Partial Test Api\*: ☒ true

Select Alg Params Api\*: ☒ true

Stop Api\*: ☒ true

Suspend Api\*: ☐ false

Parity Bit: EVEN\_PARITY ⑤

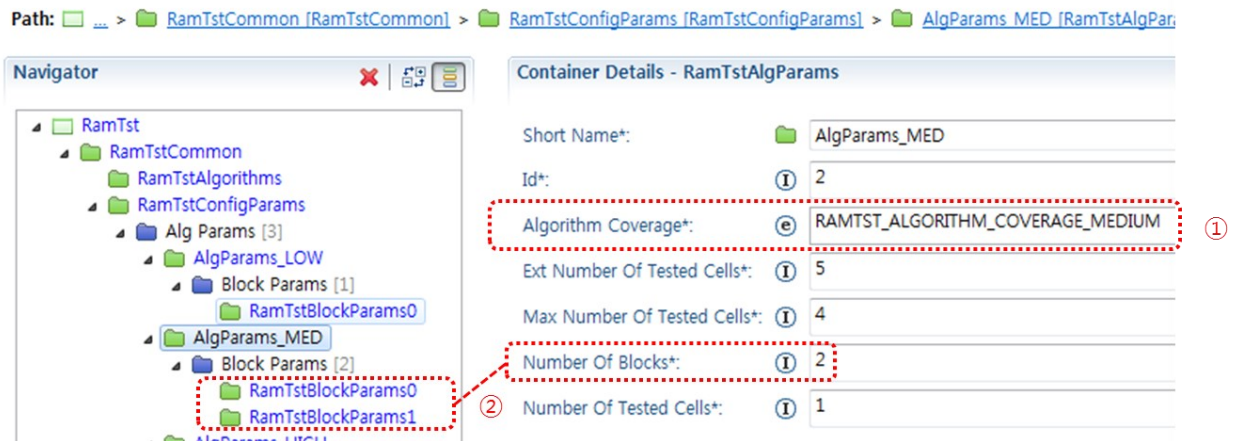
Chk Transition One To Zero: ☒ true

Crc Algorithm: RAMTST\_CRC\_8

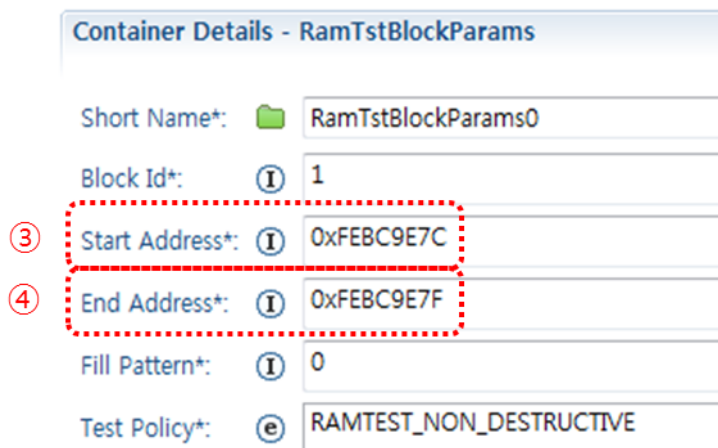
Parity Check 는 설정된 Parity Bit(⑤)에 따라 Even Parity 나 Odd Parity 를 Check 한다.

※주의: Cell 단위(예, 32 bits)로 테스트를 진행한다. 따라서 User 는 Cell 단위로 Parity 를 맞춰야 한다.

### 9.1.2 Medium Coverage Algorithm



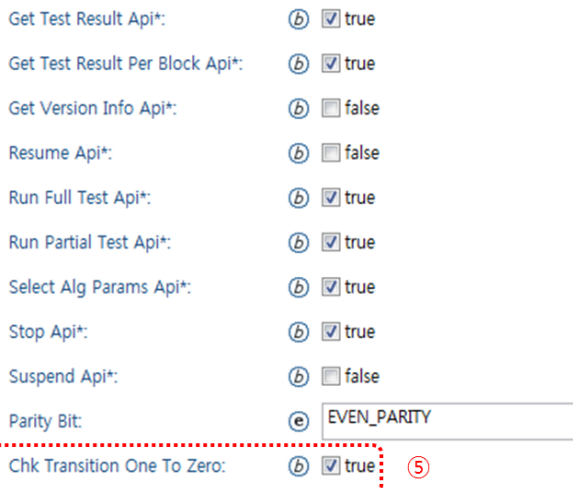
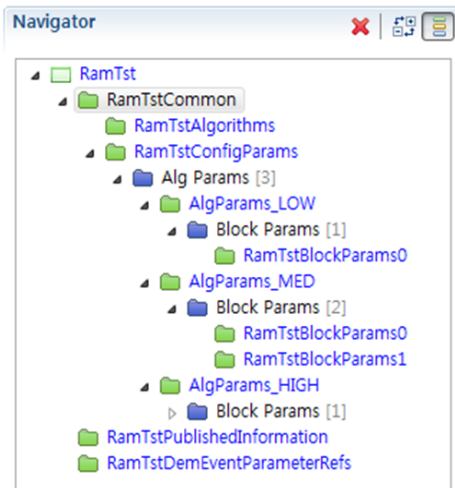
Medium Coverage Algorithm(①)은 설정된 Block 들의(②) Pattern Check 를 수행 한다.



각 Block 의 Size 는 RamTstBlockParams Container 의 Start Address(③), End Address(④)에 설정한다.

Pattern Check 는 Cell 단위로 "0x55"로 쓰고 "0x55"이 맞는지 확인후 "0xAA"로 쓰고 "0xAA"가 맞는지 확인한다.

Path: [RamTst \[RamTst\]](#) > [RamTstCommon \[RamTstCommon\]](#)



만약 Common Container 의 Chk transition one to zero(⑤)의 설정이 true 로 되어 있다면 다시 "0"을 쓰고 "0"이 맞는지를 확인한다.

Ex) Cell Size 가 8 일때, 0xAA, 0x55 로 Write / Read 확인

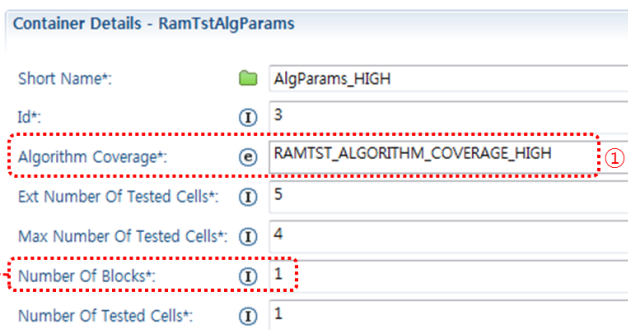
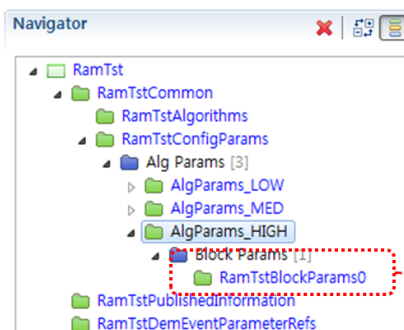
Cell Size 가 16 일때, 0xAAAA, 0x5555 로 Write / Read 확인

Cell Size 가 32 일때, 0xAAAAAAAA, 0x55555555 로 Write / Read 확인

※User 는 Parity Check 와 같은 별도의 추가 작업을 수행할 필요가 없다.

### 9.1.3 High Coverage Algorithm

Path: [...](#) > [RamTstCommon \[RamTstCommon\]](#) > [RamTstConfigParams \[RamTstConfigParams\]](#) > [AlgParams\\_HIGH \[RamTstAlgParams\]](#)



High Coverage Algorithm(①)은 설정된 Block(②)들의 CRC Check 를 수행한다.



Container Details - RamTstBlockParams

Short Name\*: RamTstBlockParams0

Block Id\*: ⓘ 1

③ Start Address\*: ⓘ 0xFEBC9E7C

④ End Address\*: ⓘ 0xFEBC9E7F

Fill Pattern\*: ⓘ 0

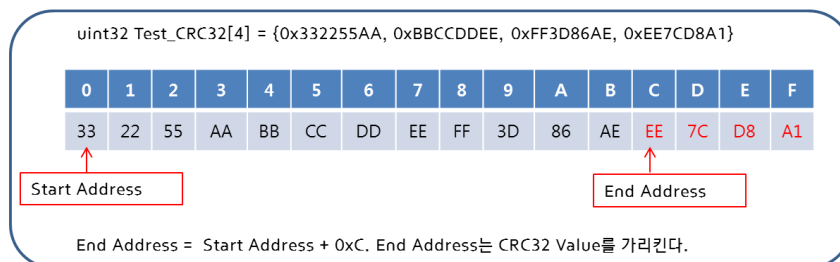
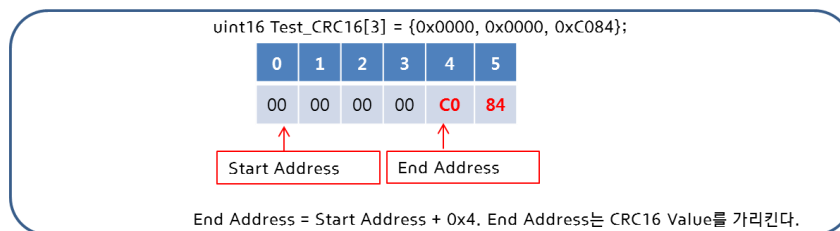
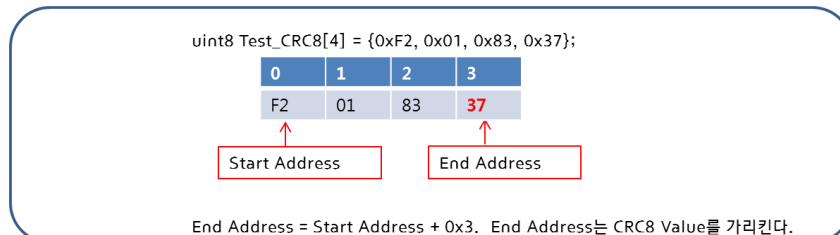
Test Policy\*: ⓘ RAMTEST\_NON\_DESTRUCTIVE

각 Block 의 Size 는 RamTstBlockParams Container 의 Start Address(③), End Address(④)에 설정한다. CRC Test 는 설정된 영역의 CRC 를 계산하여 End Address 에 쓰여져 있는 CRC 와 비교하여 값이 동일한지 확인한다.

※주의: End Address 가 CRC 가 쓰여져 있는 첫번째 byte 로 설정해야 한다. CRC8, CRC8H2F 는 1byte, CRC16 은 2byte, CRC32 는 4byte 이다.

※주의: User 는 End Address 에 CRC 값을 저장한 후 Test 를 시작해야 한다. CRC 값은 CRC 모듈의 API 를 통하여 User 가 계산할 수 있다.

Ex) Start Address : 0x0, End Address : 0x3, CRC8 를 선택했다면, 0x0 부터 0x2 까지의 CRC 를 계산하여 0x3 의 값과 비교하여 CRC 값이 동일하면 Test OK, CRC 값이 틀리면 Test NOT OK 가 된다.



## 9.2 설계시 유의사항

### 9.2.1 Foreground Test

#### 9.2.1.1 설명

"전체"와 "부분"은 전체 또는 부분 메모리를 의미하며 알고리즘의 전체 또는 부분 집합이 아니다. 테스트는 하나의 알고리즘만 사용하여 지정된 메모리 영역에 대해 수행된다. API RamTst\_SelectAlgParams 를 호출하여 알고리즘 을 선택한 후 Foreground test API 를 호출한다.

메모리 영역을 중단하지 않고 테스트할 수 있기 때문에 후 Foreground test 의 fault coverage 는 동일한 알고리즘의 background test 보다 우수하다.

※주의: RamTst\_RunFullTest, RamTst\_RunPartialTest 는 Mcu Clock 과 설정(test 범위)에 따라 시간이 오래 걸릴수 있으며, Test 내에서 ISR 까지 막는 구간이 있으므로, API 를 호출하는 Task 의 주기 및 우선순위를 잘 고려해야 한다.

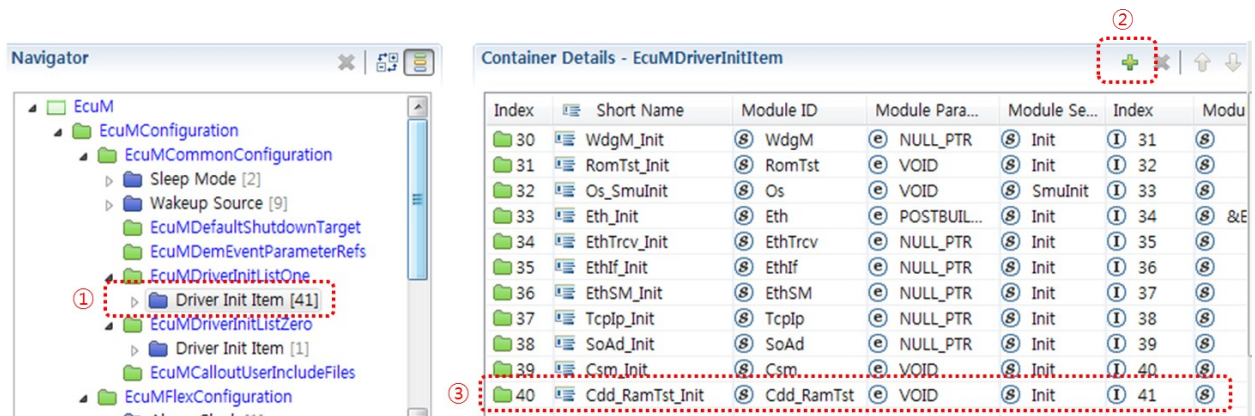
※Caution: Foreground test should not be preempted because it must be performed without interruption. Therefore, the system designer tests should be performed only at(Initialization).

#### 9.2.1.2 Sequence Diagrams

N/A

#### 9.2.1.3 설정 및 가이드

- 1) EcuM Driver Init Item 추가








- (1) EcuM/EcuMConfiguration/EcuMCommonConfiguration/EcuMDriverInitListOne/


Driver Init Item(①)을 선택한 후 +버튼을 클릭하여(②) 새로운 Item 을 생성한다(③)


## 2) EcuM Driver Init Item 설정

Container Details - EcuMDriverInitItem

Short Name*:		Cdd_RamTst_Init	①
Module ID:		Cdd_RamTst	②
Module Parameter*:		VOID	③
Module Service:		Init	④
Index*:		41	⑤

▼ To Be Configured:

Module Post Build Ptr: 

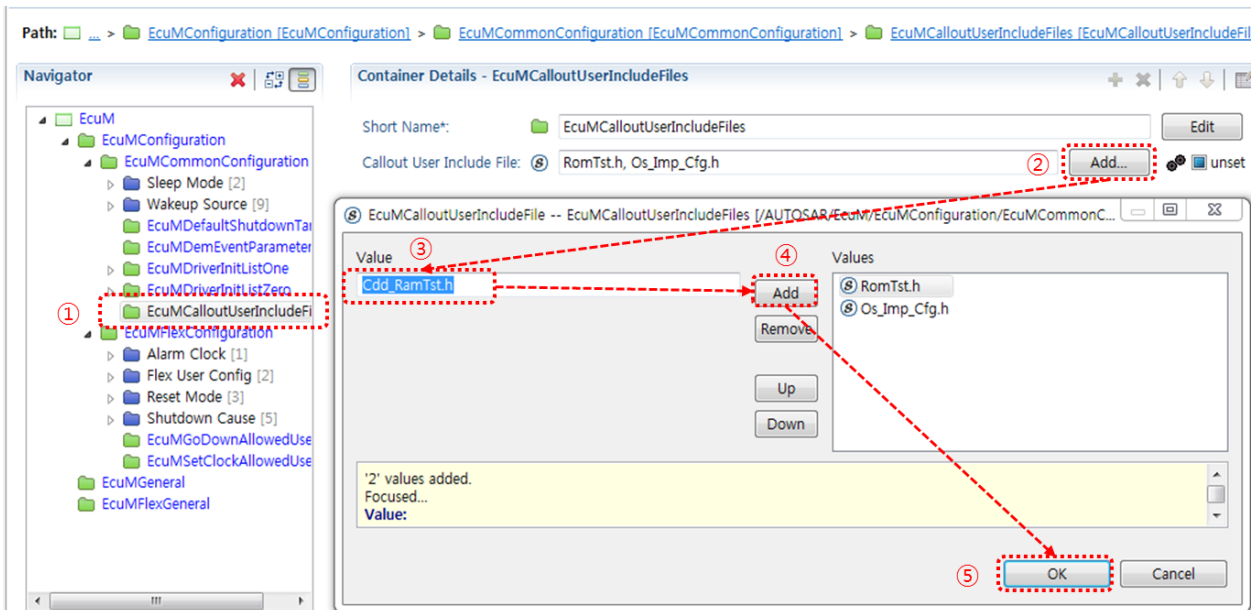
Module Ref: 

- (1) ShortName 을 Cdd\_RamTst\_Init 로 변경한다.
- (2) Module Id 에 Cdd\_RamTst 를 입력한다.
- (3) Module Parameter 를 VOID 로 선택한다.
- (4) Module Service 를 Init 로 입력한다.
- (5) Index 는 Gap 없는 순차적인 값을 입력한다.

※ 주의: Cdd\_RamTst\_Init 을 마지막에 둔 것은 이해를 돕기 위한 샘플일 뿐이다.  
제어기 상황에 맞게 적절한 위치에 설정하여야 한다. (Watchdog 등)

※ 참조: (1) 과 (2)의 Cdd\_RamTst 등의 naming 은 임의로 정한 것이므로,  
각 제어기 애플리케이션의 naming rule 에 맞게 변경해야 한다.

### 3) EcuM Callout User Include File 설정



- (1) EcuMCalloutUserIncludeFiles 를 선택한다.
- (2) Add 버튼을 클릭한다.
- (3) User header file 을 입력한다.
- (4) Add 버튼을 클릭한다.
- (5) OK 버튼을 클릭하여 헤더 파일을 등록한다.

## 4) Pseudo code

```
void Cdd_RamTst_Init(void)
{
    RamTst_SelectAlgParams(
        RAMTST_ALGORITHM_COVERAGE_LOW or
        RAMTST_ALGORITHM_COVERAGE_MEDIUM or
        RAMTST_ALGORITHM_COVERAGE_HIGH);

    RamTst_RunFullTest or RamTst_RunPartialTest(ChosenBlock);
    testResult = RamTst_GetTestResult();
    if (RAMTST_RESULT_OK == testResult)
    {
        /* do something */
    }
    else
    {
        /* do something */
    }
}
```

**Note: Limitations 참조**

## 9.2.2 Background Test

## 9.2.2.1 설명

정의된 RAM 영역의 테스트가 끝날 때까지 RamTst\_MainFunction()를 주기적으로 호출한다. RamTst\_MainFunction 가 한번 호출될 때마다 설정된 RamTstNumberOfTestedCells 의 수치 만큼 테스트 한다. 테스트 단위는 Cell 단위(예, 32bits)이다. RamTst\_MainFunction()은 하나의 cell test 가 종료되면 인 터럽트될 수 있다.

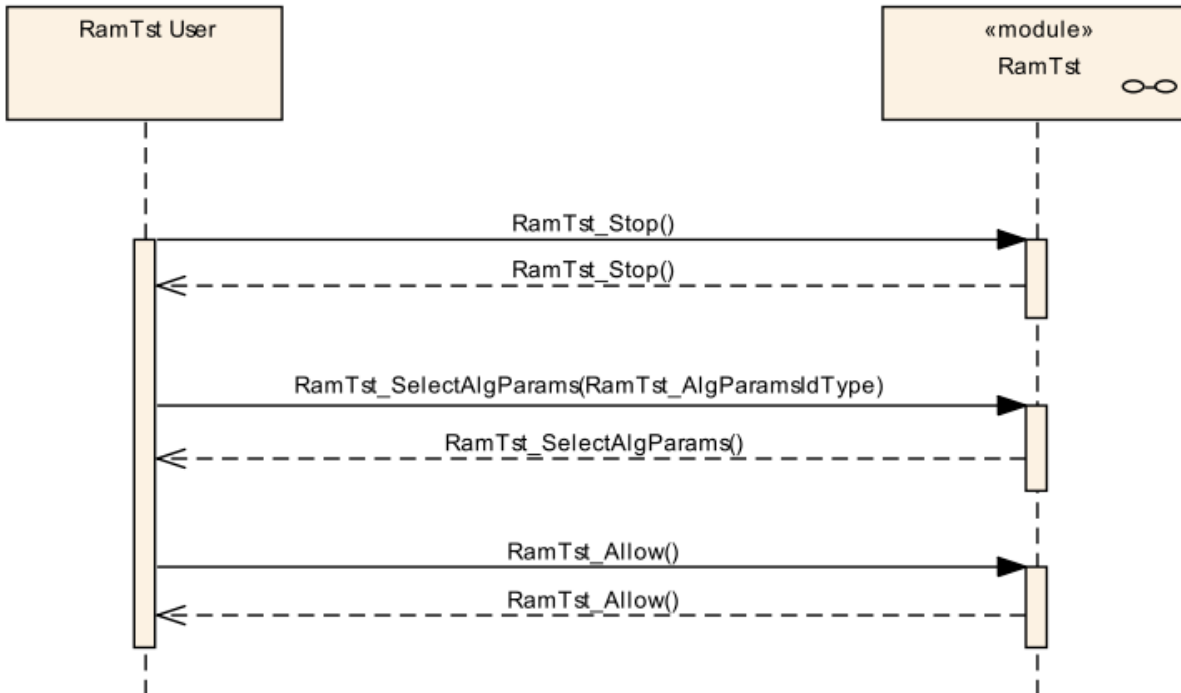
모든 block 의 테스트가 완료되면 notification callback(Test Complete Notification, Test Error Notification)을 통해 결과(OK or NOT OK)를 통보하고 처음부터 다시 테스트를 시작한다.

※주의: Application 이 초기화된 후 시작할 수 있으며 종료되기 전에 테스트가 중지되어야 한다.

※주의: 멀티코어 공유메모리의 경우, interrupt locking 을 해도 메모리 셀에 독점점으로 접근하는 것이 불가능할 수 있다. 이런 경우 공유 메모리 블록에 대한 테스트는 foreground test 를 이용해야 한다.

또한 System designer 는 독점적으로 접근 가능한 시점에 foreground test 를 수행해야 한다.

### 9.2.2.2 Sequence Diagrams



#### 1) RamTst\_SelectAlgParams

: RamTst\_SelectAlgParams 함수는 RAM 테스트 모듈에서 사용할 알고리즘을 선택한다.

참고: 이 함수를 통해 RamTstAlgParams의 구성 내용에 따라 다른 테스트 알고리즘을 선택할 수 있다. 또한 동일한 시험 알고리즘에 대해 다른 세트의 블록을 선택하는 데도 사용할 수 있다.

#### 2) RamTst\_Allow

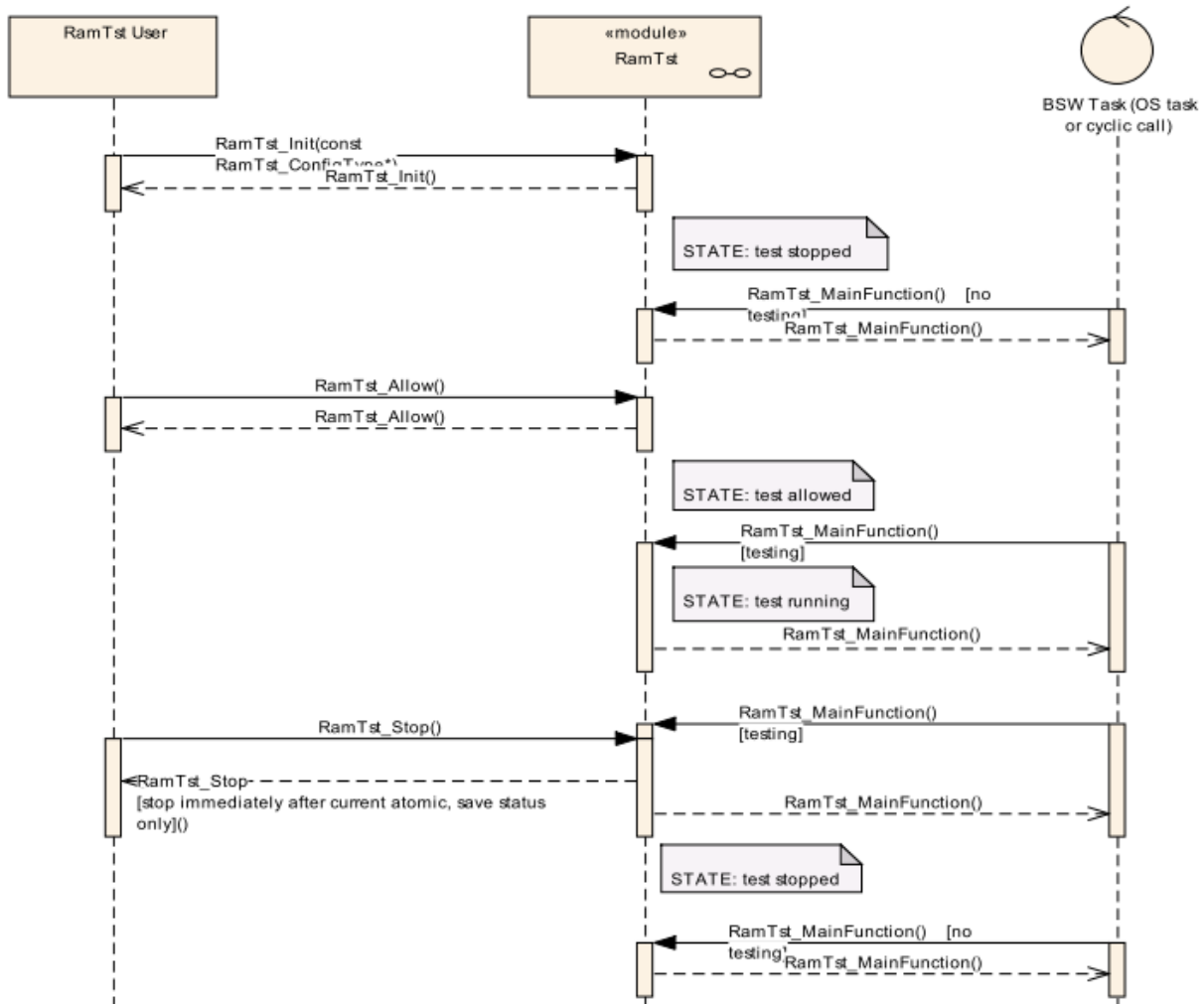
: 다음 RamTst\_MainFunction이 호출 될 때 Test가 수행되도록 상태를 변경한다.

단, RamTst가 stop 된(↓①) 상태에서 호출해야 하며, 알고리즘이 선택되지(↓②) 않았다면 default 알고리즘이(↓③) 수행된다.

① RAMTST\_EXECUTION\_STOPPED(by RamTst\_Init or RamTst\_Stop)

② RamTst\_SelectAlgParams Interface

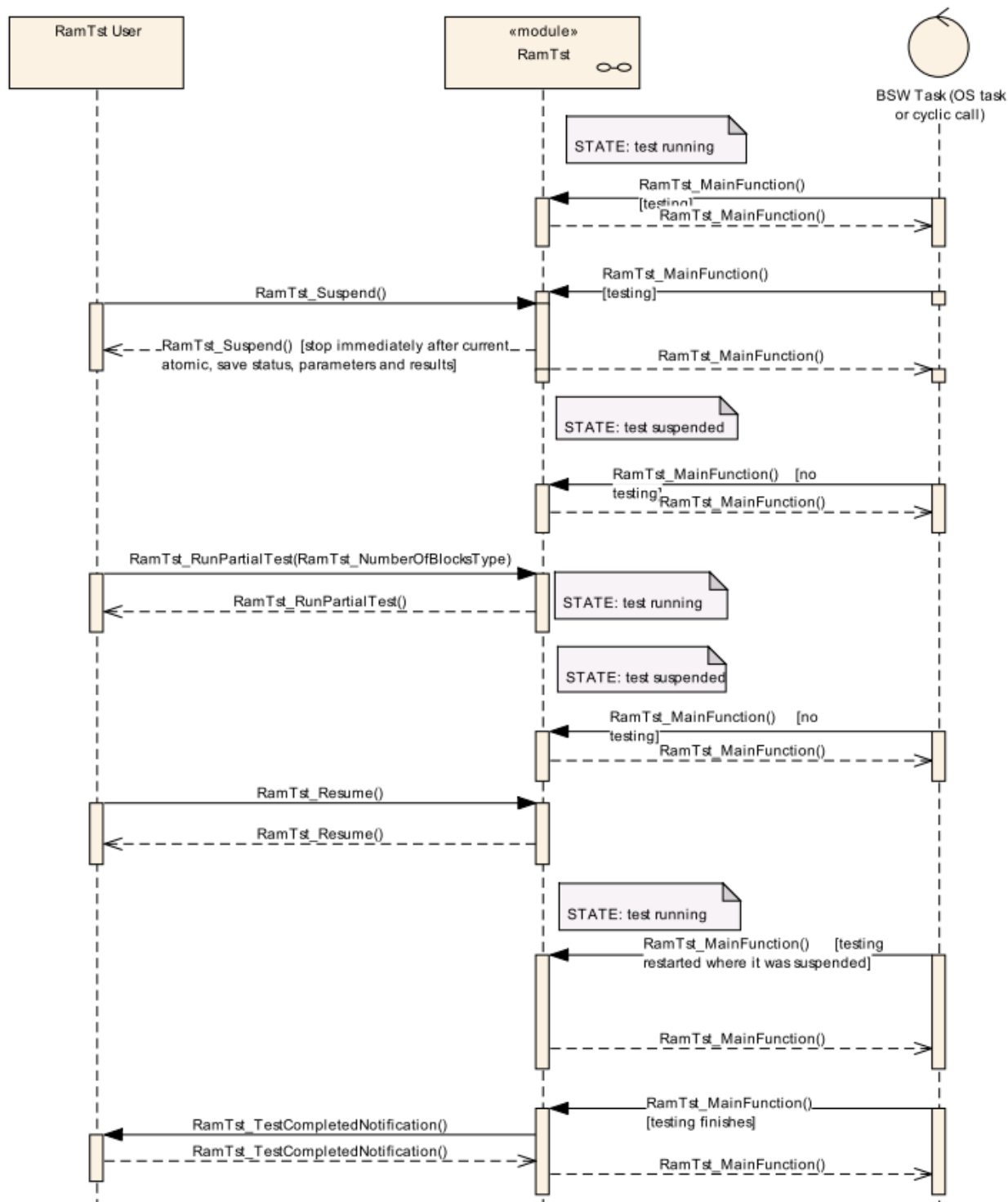
③ 설정 <RamTst/RamTstCommon/RamTstConfigParams/RamTstDefaultAlgParamsId>



### 3) RamTst\_Stop

: RamTst\_Stop 기능이 호출될 때 RamTst\_MainFunction 은 현재 실행 중인 테스트를 완료해야 하며 (RamTstNumberOfTestedCells), 이후 상태는 RAMTST\_EXECUTION\_STOPPED 로 변경된다.

시험 결과는 유지되지만 테스트를 위한 상태값들은 유지되지 않는다.



4) RamTst\_Suspend

: RamTst\_Suspend 함수는 RamTst\_MainFunction 이 다음 호출에서 테스트를 수행하지 못하도록 일시적으로 제한한다. RamTst\_Suspend 가 호출되면 실행 상태가 RAMTST\_EXECUTION\_RUNNING 에서 RAMTST\_EXECUTION\_SUSPENDED 로 변경되며 , 테스트 결과 및 현재 테스트 상태는 유지된다.



## 5) RamTst\_Resume

: RamTst\_Resume 함수는 RamTst\_MainFunction 이 중단된 지점에서 다음 테스트를 계속 진행하도록 요청한다. 테스트는 저장된 테스트 상태에서 계속 진행된다. 실행 상태를 RAMTST\_EXECUTION\_SUSPENDED 에서 RAMTST\_EXECUTION\_RUNNING 으로 변경한다.

## 6) RamTst\_TestCompletedNotification

: 모든 RAM block 의 background test 가 fail 없이 완료되었을 때 불려진다.

## 7) RamTst\_ErrorNotification

: Background test 동작 중 fail 이 발견되었을 경우 불려진다.

### 9.2.2.3 설정 및 가이드

※참조: Application Guide 문서 SAG\_RamTst.pdf 참조

### 9.2.3 RamTst\_TestResultType

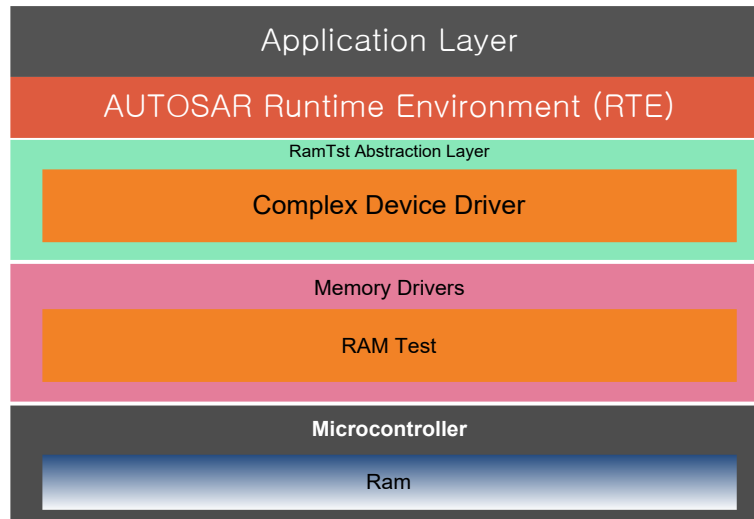
#### 9.2.3.1 RamTst\_GetTestResult(void)

RAMTST_RESULT_NOT_TESTED	if no test was started yet (after reset or deinit).
RAMTST_RESULT_OK	if all blocks have been tested with result status RAMTST_RESULT_OK.
RAMTST_RESULT_NOT_OK	if at least one block test result is RAMTST_RESULT_NOT_OK regardless whether all blocks have been already tested or not.
RAMTST_RESULT_UNDEFINED	if a test was started, not all blocks have yet been tested and no block result is RAMTST_RESULT_NOT_OK.

#### 9.2.3.2 RamTst\_GetTestResultPerBlock (block)

RAMTST_RESULT_NOT_TESTED	if this block is considered as not yet tested.
RAMTST_RESULT_OK	if all memory cells in this block have been tested sucessfully.
RAMTST_RESULT_NOT_OK	if a failure has been detected for at least one memory cell in this block
RAMTST_RESULT_UNDEFINED	if a test on this block is running.

## 9.2.4 RamTst API Call-Context



- 1) RamTst 모듈은 Mcal Layer 에 해당한다. 이에, Application 에서 RTE 를 통하여 직접 RamTst API 를 호출할 수 없다. RamTst User 가 RamTst Abstraction Layer 에 해당하는 CDD 를 만들어서 여기서 Application 과 CDD 를 연결하고, CDD 에서 RamTst API 를 호출할 수 있다.
- 2) CDD 에서는 제어기의 설계에 따라 CDD 의 API 하나당 RamTst API 하나와 1 대 1 Mapping 이 될 수도 있고, CDD 의 API 하나에 다수의 RamTst API 를 호출할 수도 있다. 이때, Chap 6.3 에 언급된 RamTst API(Init, Deinit, MainFunction 제외)들이 서로 Preemption 이 발생하지 않아야 한다. 따라서 동일한 Task 에서 RamTst API 들을 호출하도록 한다. 만약, 서로 다른 Task 에서 RamTst API 들을 호출시에는 Resource 를 동일하게 설정하여 서로 Preemption 이 발생하지 않도록 해야 한다. 또한 Low Power Mode 에서 호출해서는 안된다. 이는 High Power Mode 에서만 동작하도록 설계되어 있기 때문이다.

## 9.2.5 Low Power Mode

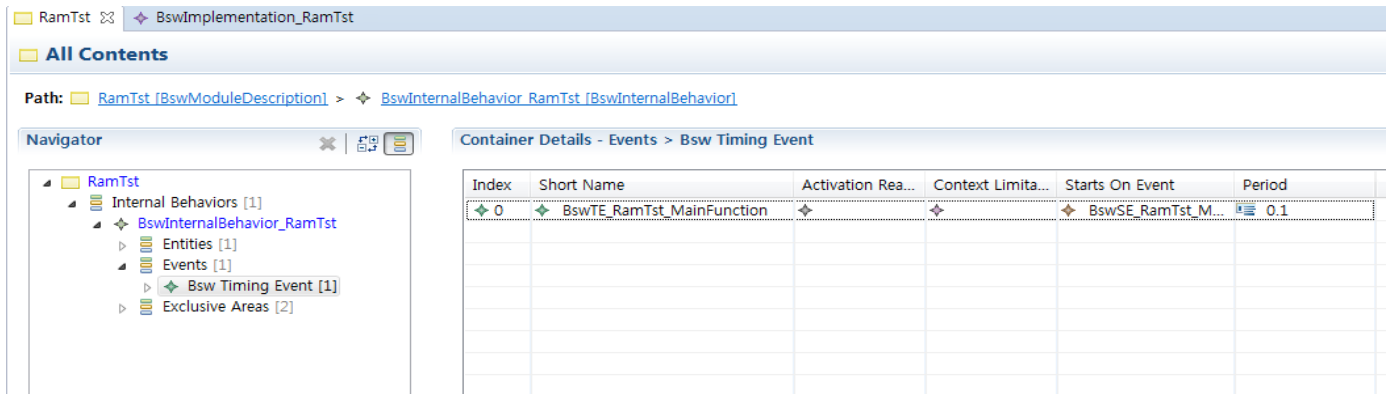
- 1) RamTst 동작 중에는 LowPower Mode 에 진입할 수 없도록 설계되어 있다. 따라서 LowPower Mode 에 진입을 할 때, 상태 값을 읽어서(RamTst\_GetExecutionStatus API) RAMTST\_EXECUTION\_STOPPED 이 아니라면, RamTst\_Stop API 를 호출해야 한다. 특히, BackGround Test 시에 suspend 상태이면 LowPower Mode 에 진입하지 않으므로 주의해야 한다.

## 9.3 Bswmd (Bsw Module Description)

### 9.3.1 MainFunction 주기 설정

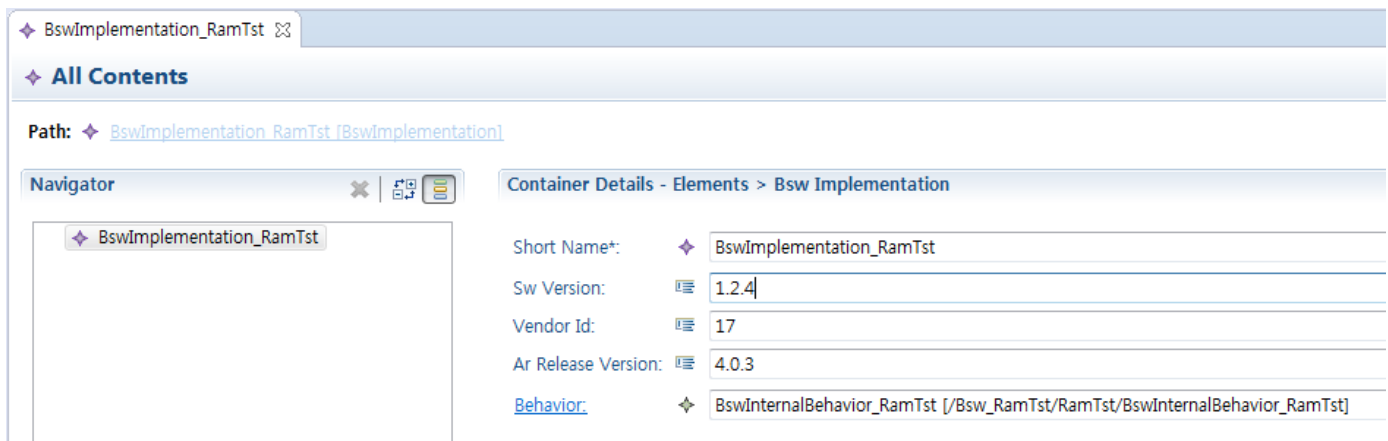
RamTst 모듈은 BackGround Test 시에 주기적으로 MainFunction 을 호출해야 하며 이를 TimingEvent 에 Mapping 하여 주기적으로 호출한다.

아래 그림과 같이 RamTst 의 BswModuleDescription Container 에 Bsw Timing Event 에서 주기를 설정한다. Period 의 단위는 second 이며 0.1 로 설정한다.



### 9.3.2 Bsw 모듈 version 설정

각 모듈을 컴파일 할 때, version 정보가 맞지 않으면 Compile 에서 Error 를 발생시킨다.  
이때는 Bswmd 의 다음과 같이 BswImplementation Container 에서 version 정보를 수정해야 한다.



## 9.4 Exclusive Areas

### 9.4.1 모듈 별 SchM Apis

In order to provide data integrity of shared resources, RamTst Module uses the scheduler service to enable and to disable data protection.

Following exclusive areas along with scheduler services are used to provide the protection:

Module	SchM APIs
RamTst	SchM_Enter_RamTst_RAM_PROTECTION() SchM_Exit_RamTst_RAM_PROTECTION() SchM_Enter_RamTst_PATTERN_CHECK_PROTECT() SchM_Exit_RamTst_PATTERN_CHECK_PROTECT()

9.4.2    설정방법

모듈 BswModuleDescription Container 의 Exclusive Areas 에 다음과 같이 추가

Navigator

RamTst

Internal Behaviors [1]

BswInternalBehavior\_RamTst

Entities [1]

Events [1]

Exclusive Areas [2]

RAM\_PROTECTION

PATTERN\_CHECK\_PROTECT

Container Details - Exclusive Areas

Index	Short Name
0	RAM_PROTECTION
1	PATTERN_CHECK_PROTECT