



SCOPE OF APPLICATION All Project/Engineering	HYUNDAI AutoEver	SHT/SHTS 1 / 39
Responsibility: 클래식오토사팀	AUTOSAR Os Profiler User Manual	DOC. NO
AUTOSAR Os Profiler User Manual		

Document Change History				
Date (YYYY-MM-DD)	Ver.	Editor	Chap	내용(개정 전 -> 개정 후)
2016-07-19	1.0.0	SH.Yoo		• Initial Creation
2016-09-02	1.0.1	SH.Yoo	3.2 4.2 5.1.1 All	<ul style="list-style-type: none"> Modified a term 'Isr' to 'Category2 Isr' Modified version information Fixed wrong reference information Fixed wrong or strange expressions
2016-09-09	1.0.2	SH.Yoo	4.3.2 5.1.1 9.1	<ul style="list-style-type: none"> Add a limitation item Delete PM configuration
2016-10-28	1.0.3	SH.Yoo	4.2 4.3.2	<ul style="list-style-type: none"> Modify a limitation item
2016-12-01	1.1.0	SH.Yoo	4.3.2 5.1.2 6.3.6	<ul style="list-style-type: none"> Modify a limitation item Modify a prototype of Opf_GetTaskInfo
2017-03-28	1.1.1	JH.Lee	4.3.1	<ul style="list-style-type: none"> Fixed warning
2017-08-29	1.1.2	MJ.Woo	4.3.2 5.1.1 5.1.2	<ul style="list-style-type: none"> Add TC2xx support
2018-03-09	1.2.0	YH.Han	4.3.2 5.1.2 6.1.3 6.1.4 8.6 8.7 8.8 8.9	<ul style="list-style-type: none"> Add SPC58xx support version at limitation Add supporting SPC58xx and TC2xx at TargetOperatingSystem Add description of Tdd_Opf_ProfileTaskInfo and Tdd_Opf_ProfileEventQ Add description of max Jitter, max CET, max GET and max RT
2018-11-19	1.3.0	YH.Han	4.3.1 4.3.2 5.1.1. 8.3 8.10	<ul style="list-style-type: none"> Add TC27x Multi-Core Support

Edition Date : 2022-08-10	File Name: Os_Profiler_UM.docx	Creation Joochan Kim	Check HoiMin Kim	Approval JiHoon Jeong
Document Management System				


	Document Name : AUTOSAR Os Profiler User Manual	Page : 2/39
---	--	------------------------------

2019-05-23	1.4.0	YH.Han	4.3.1 4.3.2 5.1.2	<ul style="list-style-type: none"> • Add S32K1xx Support
2019-11-13	1.5.0.0	YH.Han	4.3.1 4.3.2 5.1 8.10	<ul style="list-style-type: none"> • Add new configurator for AUTRON_COMMON
2020-06-17	1.5.1.0	YH.Han	4.3.1 4.3.2 5.1	<ul style="list-style-type: none"> • Change category of some configuration items • Change StmFrequency to input KHz unit • Change limitation for s32k (change default timer LPIT -> FTM)
2020-11-30	1.5.2.0	YH.Han	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.2.0
2021-01-13	1.5.3.0	MJ.Woo	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.3.0
2021-12-21	1.5.4.0	YH.Han	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.4.0
2022-02-18	1.5.5.0	JC.Kim	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.5.0
2022-03-23	1.5.6.0	YH.Han	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.6.0
2022-04-05	1.5.7.0	YH.Han	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.7.0
2022-07-04	1.5.7.1	HJ.Kim	1 4.3.1	<ul style="list-style-type: none"> • Clarify copyright of the code • Release OsProfiler-1.5.7.1
2022-08-10	1.5.8.0	JC.Kim	4.3.1	<ul style="list-style-type: none"> • Release OsProfiler-1.5.8.0


	Document Name : AUTOSAR Os Profiler User Manual	Page : 3/39
---	--	------------------------------

목차

1. OVERVIEW	5
2. REFERENCE	5
3. AUTOSAR SYSTEM	6
3.1 OVERVIEW OF SOFTWARE LAYERS	6
3.2 OS PROFILER	7
4. PRODUCT RELEASE NOTES.....	9
4.1 OVERVIEW	9
4.2 SCOPE OF THE RELEASE.....	9
4.3 MODULE RELEASE NOTES	9
4.3.1 Change Log	9
4.3.2 Limitation	17
4.3.3 Deviation	18
5. CONFIGURATION GUIDE.....	19
5.1 OS PROFILER 모듈	19
5.1.1 OsProfilerGlobalConfig Container	19
5.1.2 OsProfilerSpecificTargetConfig Container	21
5.1.3 OsProfilerCommonTargetConfig Container.....	21
5.1.4 OsProfilerPeriodTaskConfig Container.....	21
6. APPLICATION PROGRAMMING INTERFACE (API).....	23
6.1 TYPE DEFINITIONS	23
6.1.1 OpfRecModeType.....	23
6.1.2 OpfTaskIDType	23
6.1.3 Tdd_Opf_ProfileTaskInfo.....	23
6.1.4 Tdd_Opf_ProfileEventQ	23
6.2 MACRO CONSTANTS.....	24
6.3 FUNCTIONS	24
6.3.1 Opf_InitOsProfiler	24
6.3.2 Opf_StartOsProfiler	24
6.3.3 Opf_StopOsProfiler	25
6.3.4 Opf_RestartOsProfiler	25
6.3.5 Opf_GetOsProfilerRecMode	26

	Document Name : AUTOSAR Os Profiler User Manual	Page : 4/39
---	--	------------------------------

6.3.6	Opf_GetTaskInfo	26
6.3.7	Opf_GetProfileInfo	26
7.	GENERATOR.....	28
7.1	GENERATOR OPTION	28
8.	OS PROFILER.....	29
8.1	동작 모드	29
8.2	SYSTEM TIMER CLOCK	29
8.3	이벤트 QUEUE	30
8.4	PROFILE 이벤트 QUEUE	32
8.5	SLACK TIME	32
8.6	JITTER.....	34
8.7	CET (CORE EXECUTION TIME)	35
8.8	GET (GROSS EXECUTION TIME).....	36
8.9	RT (RESPONSE TIME)	37
8.10	MULTI-CORE 지원	38
9.	APPENDIX.....	39
9.1	설계시 유의사항.....	39
9.2	EXCLUSIVE AREAS	39

	Document Name : AUTOSAR Os Profiler User Manual	Page : 5/39
---	--	----------------

1. Overview

본 문서는 OS Profiler SRS 를 기반으로 작성되었으며 OS Profiler 모듈 사용시 보다 자세한 기능적인 설명이 필요한 경우 Reference 문서를 참조해야 한다.

설정관련 Category의 해석은 다음과 같다.

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

This source code is permitted to be used only in projects contracted with Hyundai Autoever, and any other use is prohibited.

If you use it for other purposes or change the source code, you may take legal responsibility.

In this case, There is no warranty and technical support.

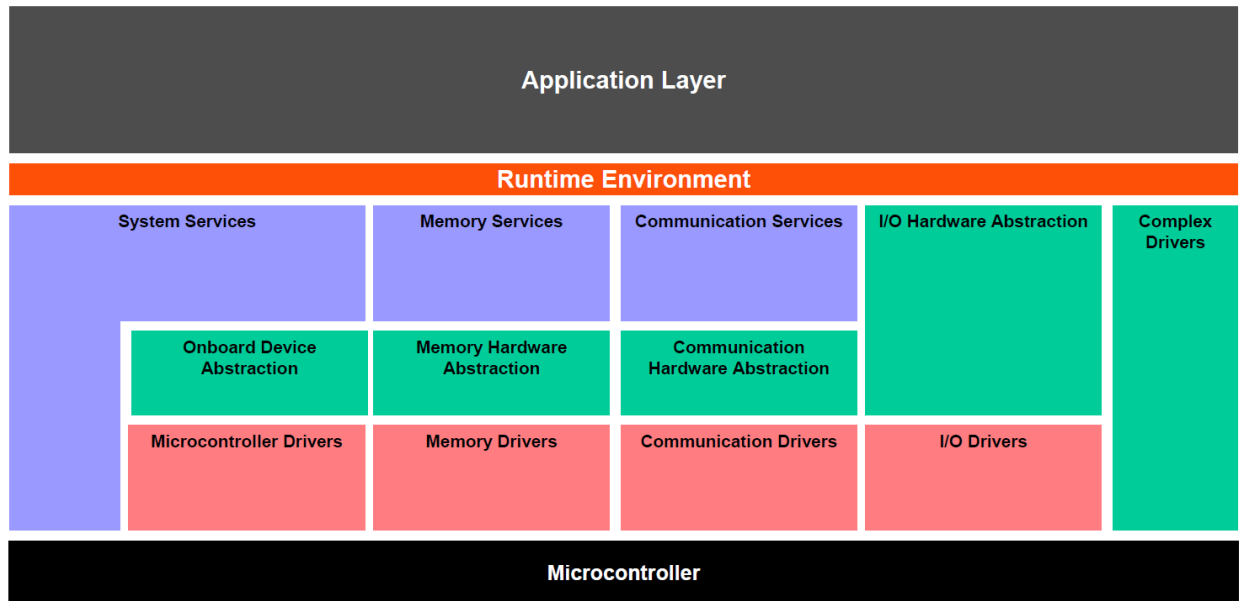
2. Reference

Sl. No.	Title	Version
1.	Os_Profiler_ESRS.docx	1.5.0

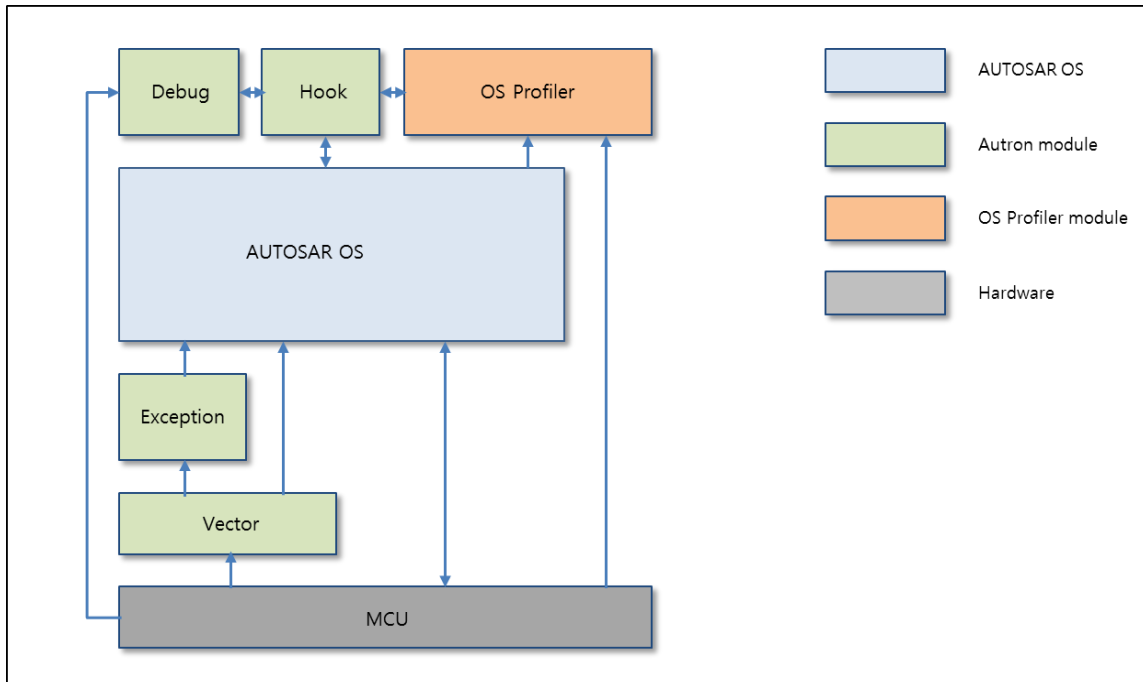
3. AUTOSAR System

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.



AUTOSAR OS 는 System Service 에 속하며, 하기 그림과 같은 구조를 가진다. AUTOSAR OS 를 기반으로 Vector, Exception, Debug, Hook 의 AutoEver 모듈이 추가되어 전체적인 AUTORSAR OS 모듈을 완성한다. 그림에서 표현된 OS Profiler 는 AUTOSAR OS 를 기반으로 Task/Category2 ISR 의 이벤트를 수집한다.



3.2 OS Profiler

OS Profiler 모듈은 OS 에서 동작하는 주기Task 및 Category2 Isr 의 상태 정보를 실시간으로 수집하고, 이를 기반으로 사용자는 Task 나 Category2 Isr 의 동작을 파악할 수 있다. 특히, 주기 Task 이전의 다른 Task 및 Category2 Isr 정보를 기반으로 해당 주기 Task 가 정확한 시점에 Running* 상태에 진입할 수 있는지를 추정할 수 있다.

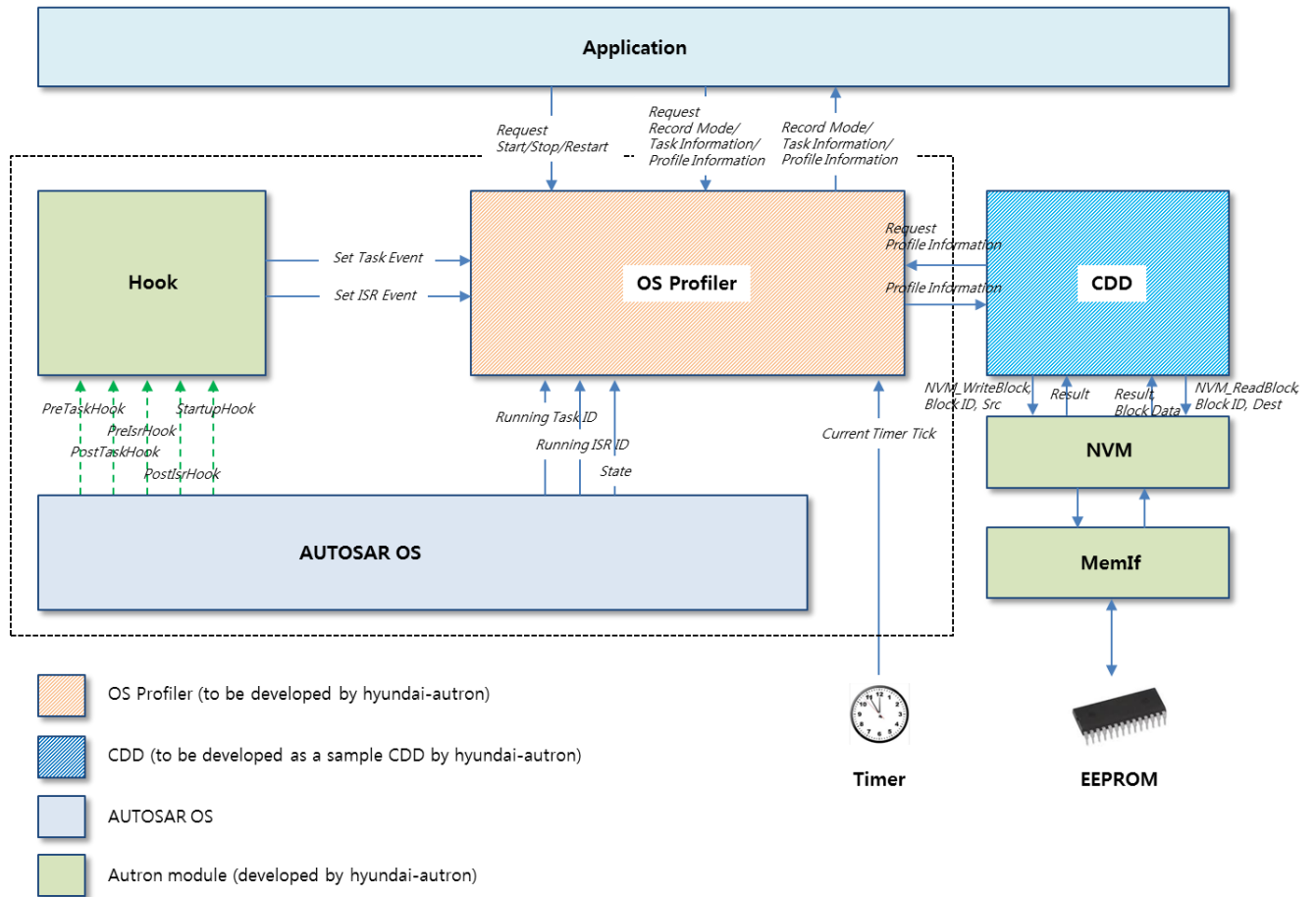
정리하면, OS Profiler 모듈이 수행하는 주요 기능은 다음과 같다.

- 실시간으로 Task 및 Category2 Isr 의 상태 정보 수집
- 주기 Task 이전의 MCU Core 의 '유휴 상태**' 정보를 수집

* Running : Task 의 상태 중 하나. Task 가 MCU 자원을 선점하여 실제로 동작하고 있는 상태를 의미한다.

** 유휴 상태 : Idle 상태라고도 하며, MCU Core 가 아무런 일도 하지 않는 상태를 의미한다. 이 상태에서는 Task 나 Isr 이 바로 실행(Task의 경우, Running 상태로 천이)할 수 있다.

OS Profiler 모듈을 중심으로, Application, Hook, AUTOSAR OS, CDD 의 Software 모듈과 연결되어 있고, Hardware 모듈로는 Timer 와 연결되어 있다.



4. Product Release Notes

4.1 Overview

이 Chapter에서는, 현대오트오에버 OS profiler에 대한 release 관련 내용을 제공하는데 목적이 있으며, OS Profiler Software product release version에 대한, 제한사항 및 특이사항을 기술하고 있다.

4.2 Scope of the release

이 문서에 대한 모든 내용은, 다음의 현대오트오에버 OS Profiler 모듈에 한정한다.

Module	Module version
OS Profiler	1.5.8

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

4.3 Module release notes

4.3.1 Change Log

4.3.1.1 Version 1.5.8.0

➤ 개선 사항

■ UNECE Cyber Security 범규 대응 보안 코딩 개선

원인	고객 요청
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.2 Version 1.5.7.1

➤ 개선 사항

■ Clarify the copyright of code in E-code, Generate Code

Cause	Change request about copyright
Operation Impact	N/A
Configuration Impact	N/A
Required measure of ASW	N/A

■ Content file DeliveryBoxHistory modification

Cause	Change request to apply new template
Operation Impact	N/A
Configuration Impact	N/A

Required measure of ASW	N/A
-------------------------	-----

4.3.1.3 Version 1.5.7.0

➤ 개선 사항

■ EnableOsProfiler 설정 false 시의 link error 해결

원인	OsProfiler-1.5.4.0 에서의 폴더 구조 변경으로 해당 설정 false 시에 link error 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.4 Version 1.5.6.0

➤ 개선 사항

■ UNECE Cyber Security 법규 대응 보안 코딩 개선

원인	UNECE Cyber Security 법규 위반 사항 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ generator에서 파일 생성 시에 input file arxml 파일 목록 주석이 정렬되어 출력되도록 변경

원인	매 생성시 마다 입력 arxml 파일 순서가 변경되는 불편사항 개선
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.5 Version 1.5.5.0

➤ 개선 사항

■ Linking 시 일부 변수 잘못된 Memory Section 지정 오류 수정

원인	일부 변수의 Memory Section 지정 오류
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ RH850 Object ID type mismatching 으로 인한 E_OS_ID ErrorHook 발생하는 오류 수정

원인	ObjectID 관련 변수의 잘못된 Type 선언 수정
동작 영향	E_OS_ID 에러 미발생
설정 영향	없음
ASW 조치 필요 사항	없음

■ Bolero API extern 선언문 문법 오류로 build error 발생하는 오류 수정

원인	Bolero MPC560XB MCU API extern 선언문에서 input parameter 의 type 이 누락
동작 영향	Build Error (error #92) 미발생

설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.6 Version 1.5.4.0

➤ 개선 사항

■ UNECE Cyber Security 법규 대응 보안 코딩 개선

원인	UNECE Cyber Security 법규 위반 사항 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ 신규 문서 템플릿 적용

원인	회사 합병으로 인한 템플릿 변경
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.7 Version 1.5.3.0

➤ 개선 사항

■ 정적 검증 위반 사항 수정 또는 정당화

원인	정적 검증 결과 반영 필요
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.8 Version 1.5.2.0

➤ 개선 사항

■ Generator 에서 8 진수 입력 허용

원인	8/10/16 진수 입력 허용 가능한 파라미터들에 대해서 8 진수 입력 시 잘못된 값으로 인식
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ 정적 검증 위반 사항 수정 또는 정당화

원인	정적 검증 수행 필요
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.9 Version 1.5.1.0

➤ 개선 사항

■ OsProfiler StmFrequency 설정 값 KHz 입력 허용 (소수점 이하 자리 통해 KHz 입력)

원인	일부 MCU 는 KHz 값의 타이머 frequency 제공하여 기존 MHz 단위 입력으로 처리 불가
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ 일부 항목 category 변경

원인	MCU 의존적인 항목에 대해서 category 변경
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.10 Version 1.5.0.0

➤ 신규 기능

■ AUTRON_COMMON OsProfiler 설정 신규 추가

원인	OsProfiler 에 대한 MCU 의존성 제거
동작 영향	없음
설정 영향	MCU 에 따라 OsProfilerGlobalConfig, OsProfilerCommonTargetConfig, OsProfilerSpecificTargetConfig 설정 필요
ASW 조치 필요 사항	없음

■ Aurix TC3xx 6 core 지원

원인	Aurix TC3xx 6 Core 에 대한 OS Profiler 지원 필요
동작 영향	없음
설정 영향	OsProfiler 가 동작하는 Core 의 개수에 따라 OsProfilerGlobalConfig > NumberOfCores 설정 필요
ASW 조치 필요 사항	없음

➤ 개선 사항

■ OsProfiler TimeIndex 반복문에 대한 count break 추가

원인	TimeIndex 처리에 대해 불필요하게 오래 동안 반복문에 머무르는 경우 예외 처리
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ 설정 category 변경

원인	Category 의 Fixed 항목을 Changeable 로 변경
동작 영향	없음
설정 영향	없음

ASW 조치 필요 사항	없음
--------------	----

4.3.1.11 Version 1.4.0

➤ 신규 기능

■ S32K1xx MCU 지원

원인	S32K1XX MCU 용 OS Profiler 지원 요구
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

➤ 개선 사항

■ 컴파일 warning 제거

원인	Multicore 미지원 MCU 에서 컴파일 시에 수행되지 않는 code 로 인해 컴파일 warning 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

➤ 개선 사항

■ 정적 검증 위반 사항 수정

원인	사용자 환경 설정 등에 의한 unreachable code 등의 정적 검증 위반 사항 수정
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.12 Version 1.3.0

➤ 신규 기능

■ Aurix Multi-Core 지원

원인	Aurix Multi-Core OS Profiler 지원 요구
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.13 Version 1.2.0

➤ 신규 기능

■ ST SPC58xx MCU 지원

원인	SPC58xx MCU 용 OS Profiler 지원 요구
----	---------------------------------

동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

➤ 개선 사항

■ Task 의 period 값이 비정상적으로 표기되는 문제 수정

원인	OsProfiler 1.1.2 에서 Opf_SetPrelsrEvent 수정하면서 Opf_SaveTimeStamp() 함수가 OPF_PROFILE_ISR == STD_OFF 시, 누락됨
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ UM 문서 내용 추가

원인	기존 문서에서 오류 및 부족한 내용 아래와 같이 추가 - 5.1.2: TatgetOperatingSystem 에 TC2xx 추가 - 6.1.3, 6.1.4: Tdd_Opf_ProfileTaskInfo and Tdd_Opf_ProfileEventQ 에 대한 설명 추가 - 8.6 ~ 8.9: max jitter, max CET, max GET, max RT 에 대한 설명 추가
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.14 Version 1.1.2

➤ 신규 기능

■ Infineon TC2xx MCU 지원

원인	TC2xx MCU 용 OS Profiler 지원 요구
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

➤ 개선 사항

■ TaskInfo 정보가 다른 Task 로 잘못 보여지는 문제 수정

원인	OsProfilerPeriodTaskConfig 를 10 개 이상 설정했을 때, Generator 에 의해 생성되는 Opf_GaaProfileTaskInfo[x] 와 Opf_GaaTaskInfo[x] 간의 연결이 잘못 정렬된 채 생성
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.15 Version 1.1.1

➤ 개선 사항

■ Warning 개선

원인	불필요한 연산 수행으로 인하여 경고 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.16 Version 1.1.0

➤ 신규 기능

■ N/A

➤ 개선 사항

■ Floating 연산대신 정수 연산하도록 개선

원인	최소 유티 마진 및 Jitter 계산 시, Floating 연산 대신 정수 연산하도록 수정 (Floating 연산을 위한 추가 라이브러리 불필요)
동작 영향	최소 유티 마진 및 Jitter 가 백분율이 아닌, 천분율로 기록
설정 영향	없음
ASW 조치 필요 사항	최소 유티 마진 및 Jitter 가 정수값 천분율로 기록되므로, 해당 값을 이용하는 ASW 의 경우에는 이에 맞도록 수정 필요

■ Library 배포와 관련된 문제 개선

원인	Library 배포 당시와 사용자 사용 당시의 설정 옵션 차이로 인한 문제
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ Task Activation 확인 시점 개선

원인	기존 Counter 처리 후에서 처리 전으로 Task Activation 확인 시점 변경 필요
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ Opf_GetTaskInfo 함수 개선

원인	기존 OS 의 ID 를 그대로 인자로 사용할 수 없었던 문제 개선
----	--------------------------------------

동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	Opf_GetTaskInfo 의 함수 인자로 OS 에서 제공하는 Task ID 를 그대로 사용 가능 (별도의 변환 필요 없음)

■ RH850 에서 E_OS_LIMIT 이벤트가 기록되지 않는 문제 개선

원인	RH850 에서 E_OS_LIMIT 이벤트가 기록되지 않는 문제 발생
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

■ RH850 에서 지연된 Task 상태 정보가 갱신되지 않는 문제

원인	PostTaskHook 이후, 다른 Task 로 Context-Switch 가 바로 일어날 때, Interrupt 가 도중에 발생하는 경우에는 PreTaskHook 이 아닌 PreIsrHook 이 호출됨
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.17 Version 1.0.3

➤ 신규 기능

■ N/A

➤ 개선 사항

■ RH850 에서의 PLL m-factor 값 범위 측정 문제 개선

원인	RH850 에서 PLL m-factor 값 범위가 잘못 계산되고, 이로 인해 Timer Clock 이 정상적으로 측정되지 않음
동작 영향	특정 m-factor 값이 사용될 경우, Timer Clock 정상적으로 측정되지 않음
설정 영향	없음
ASW 조치 필요 사항	없음

■ User Manual 업데이트

원인	RH850 OS 에서의 ISR 이벤트 미지원 표기 (Chapter 4.3.2)
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.1.18 Version 1.0.2

➤ 신규 기능

■ N/A


➤ 개선 사항

■ User Manual 업데이트

원인	양산 관련 Limitation 추가 및 PM 설정 내용 삭제 필요 (Chapter 4.3.2, 5.1.1, 9.1)
동작 영향	없음
설정 영향	없음
ASW 조치 필요 사항	없음

4.3.2 Limitation

- OS Profiler 는 AUTOSAR OS 상에서 동작하기 때문에 AUTOSAR OS 의존적이다. 현재 OS Profiler 가 지원하는 AUTOSAR OS 는 하기와 같다.
 - Freescale AUTOSAR OS/MPC5600 v.4.0
 - RX-AUTOSAR850 V2, RV850 V2.01.03
 - AUTOEVER AUTOSAR OS TC2xx v2.4.0 이상
 - AUTOEVER AUTOSAR OS TC3xx v0.3.0 이상
 - AUTOEVER AUTOSAR OS SPC58xx v2.2.0 이상
 - AUTOEVER AUTOSAR OS S32K1xx v1.2.0 이상
 - AUTOEVER AUTOSAR OS S32K1xx v1.2.0 이상
 - AUTOEVER AUTOSAR OS CYTxxx v1.1.0 이상
- OS Profiler 에서 수집하는 ISR 이벤트 정보 (ISR 진입, ISR 종료) 는 Category2 ISR 로 제한되며, Category1 ISR 이벤트는 OS Profiler 에 의해 수집되지 않는다.
- OS Profiler 의 Profiling 대상 Task 는 주기 Task 로 제한된다. 이는 Profiling 정보를 수집하기 위해서는 Task 의 activation 시점을 파악해야 하는데, alarm 이 아닌 방법(ActivateTask API 등) 에 의해 activate 되는 task 들의 activation 시점을 파악하기가 불가능하기 때문이다.
- OS 의 status 는 EXTENDED status 로 설정되어야 한다.
- Multicore 환경을 부분적으로 지원한다. 현재 Aurix 에서 Multicore OS Profiler 기능을 지원하며 TC2xx 는 최대 3 개, TC3xx 는 최대 6 개의 Core 를 지원한다. 각 Core 의 동작은 동일한 STM frequency 로 동기화가 이루어진 환경에서 OS Profiler 가 정상 동작한다.
- Freescale OS 환경에서는 System Timer 를 기반으로 한, HWCOUNTER 및 SWCOUNTER 만 지

	Document Name : AUTOSAR Os Profiler User Manual	Page : 18/39
---	--	-----------------

원 가능하다. (Second Timer 및 별도 사용자가 정의한 S/W counter 지원 불가)

7. 본 기능은 최종 양산 S/W 에는 포함되지 말아야 한다.
(OS Profiler Configuration 에서 EnableOsProfiler 항목을 false 로 설정)
8. S32K1xx 는 FTM 을 기반으로 하여 OS Profiler 기능을 지원한다. FTM 은 16 비트 카운터로 표현할 수 있는 시간이 제약적이다. 따라서 OsProfiler 의 Time Index Period 설정 시에 Stm Frequency 를 고려하여 표현 $2^{16} / \text{StmFrequency}$ 이내로 period 를 결정해야 한다. 또한 OsProfiler 에서 측정 가능한 주기 task 의 period 로 $2^{16} / \text{StmFrequency}$ 로 제한된다.
9. S32K1xx 는 FTM 의 OsProfiler 는 FTM3 을 기본적으로 사용하며 해당 유닛은 internal / external watchdog 과 함께 사용한다. 따라서 OsProfiler 또는 internal / external watchdog 에서 FTM3 에 대한 설정 변경 시에 다른 모듈에 영향을 미칠 수 있다.

4.3.3 Deviation

None

5. Configuration Guide

5.1 OS Profiler 모듈

5.1.1 OsProfilerGlobalConfig Container

이 Container에서는 OS Profiler 동작에 필요한 일반적인 기능을 설정한다.

Parameter Name	Value	Category
EnableOsProfiler	true/false	C
EnableProfileIsr	true/false	C
NumberOfCores	User Defined	C
ProfileEventCount	User Defined	C
EnableTimeIndex	true/false	C
UpwardDirectionCount	true/false	F
TimeIndexPeriod	User Defined	C
TargetOperatingSystem	User Defined	F

1. EnableOsProfiler

- OS Profiler 기능을 활성화/비활성화 한다.
 - true : OS Profiler 기능 활성화
 - false : OS Profiler 기능 비활성화
- 최종 양산 S/W에서는 Os Profiler가 포함되지 않으므로, 이 항목이 false로 설정되어야 한다.

2. EnableProfilerIsr

- Category2 ISR 이벤트 수집 여부를 결정한다.
 - true : Category2 ISR 이벤트 수집
 - false: Category2 ISR 이벤트 수집하지 않음

3. NumberOfCores

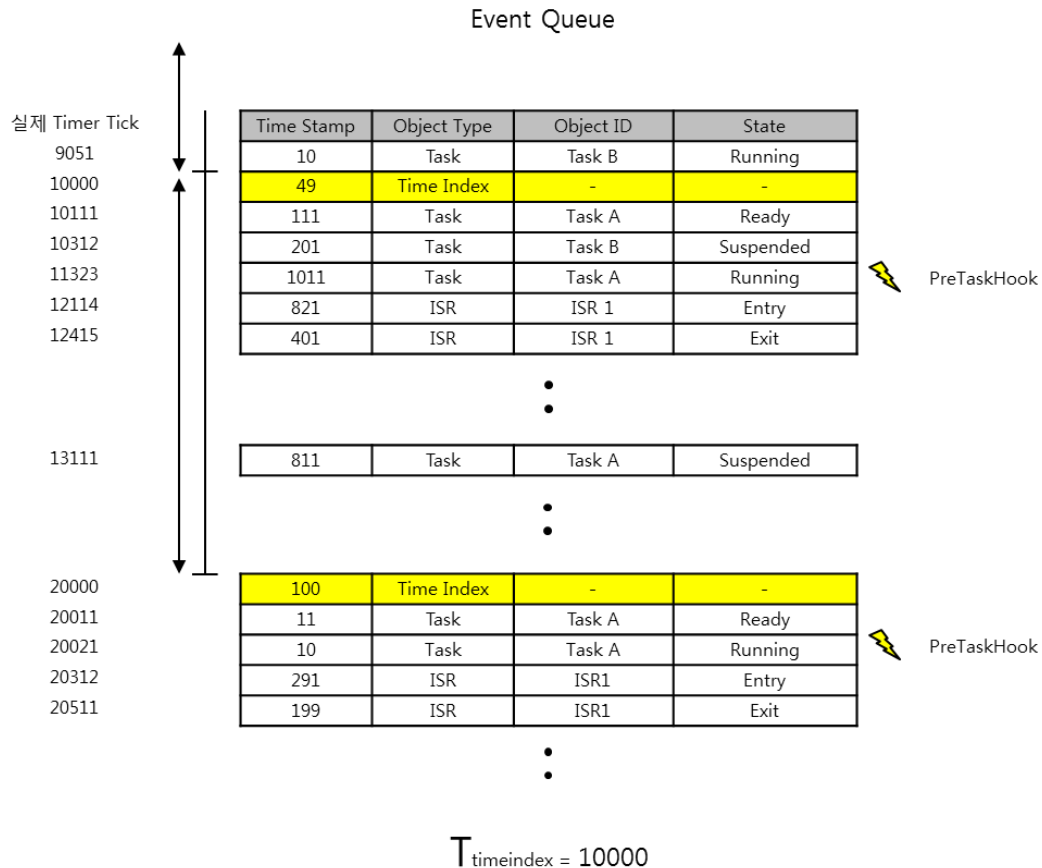
- OsProfiler가 동작하는 Core의 개수를 설정한다. TC2xx는 1~3, TC3xx는 1~6으로 설정할 수 있다. 그 외의 MCU는 오직 1로 설정 가능하다. 동작 Core는 낮은 번호의 Core부터 동작한다. 즉, 5로 설정 시에 Core0 ~ Core4에서 OsProfiler가 동작한다.

4. ProfileEventCount

- Profiling하려는 Task/Cat2ISR의 이벤트 수를 설정한다. OS Profiler에서 설정 가능한 Profile 이벤트수의 최대 값은, 32767개이다. 이벤트당 24byte (이벤트 Queue 12byte, Profile 이벤트 Queue 12byte)의 메모리 공간을 차지하기 때문에, 최대 Profile 이벤트 수는 시스템의 메모리가 허용 가능한 범위 내에서 설정되어야 한다.

5. EnableTimeIndex

- TimeIndex 이벤트 수집 여부를 결정한다. 하기와 같이, TimeIndex 는 사용자가 설정한 주기 (Period)대로 이벤트 Queue 에 저장된다.



- true : TimeIndex 이벤트 수집
- false: TimeIndex 이벤트 수집하지 않음

6. UpwardDirectionCount

- OsProfiler 가 참조하는 counter 의 동작 방향을 설정한다.
 - true : counter 가 0 에서부터 증가하는 오름 순서
 - false : counter 가 max 값에서 0 으로 감소하는 내림 순서

7. TimeIndexPeriod

- TimeIndex 이벤트의 주기(Period)를 설정한다.

8. TargetOperatingSystem

- 현재 system 이 사용하는 OS 를 설정한다. Freescale 사의 mpc560xx 를 사용하는 경우에는 FREESCALE_MPC560xB, Renesas 사의 RH850 을 사용하는 경우에는 RENESAS_RH850 으로 설정한다. 그 외에는 AUTRON_COMMON 을 선택한다.

5.1.2 OsProfilerSpecificTargetConfig Container

이 Container는 TargetOperatingSystem이 FREESCALE_MPC560xB 또는 RENESAS_RH850 일 경우에 참조한다.

Parameter Name	Value	Category
XOSC	User Defined	C

1. XOSC

- ECU 에서 사용되는 외부 Oscillator Clock 을 설정한다. 이 값은 OS Profiler 가 사용하는, System Timer Clock 을 계산하기 위해 사용된다.

5.1.3 OsProfilerCommonTargetConfig Container

이 Container는 TargetOperatingSystem이 AUTRON_COMMON 일 경우에 참조한다.

Parameter Name	Value	Category
StmAddress	User Defined	F
StmMaxTick	User Defined	F
StmFrequency	User Defined	F

1. StmAddress

- OsProfiler 가 참조하는 timer 주소를 설정한다.

2. StmMaxTick

- OsProfiler 가 참조하는 timer 의 최대 표현 가능한 값을 설정한다.


3. StmFrequency

- OsProfiler 가 참조하는 timer 의 동작 주파수 값을 MHz 단위로 설정한다. 소수점을 통해 KHz 단위 입력 가능하다.

5.1.4 OsProfilerPeriodTaskConfig Container

이 Container 에서는 OS Profiler 가 Profiling 할, 주기 Task 를 설정한다.

Parameter Name	Value	Category
PeriodTaskRef	User Defined	C
TaskPeriod	User Defined	N

	Document Name : AUTOSAR Os Profiler User Manual	Page : 22/39
---	--	-----------------

1. PeriodTaskRef

- Profiling 할 Task 참조를 설정한다. 참조할 Task 는 OS 에 이미 설정된 주기 Task 가 대상이 된다.

2. TaskPeriod

- Profiling 할 Task 의 주기를 초 단위로 설정한다. 이 값은 현재 사용되지 않는다.

6. Application Programming Interface (API)

6.1 Type Definitions

6.1.1 OpfRecModeType

Type:	Scalar
Range	0 ~ 3
Description:	Data type for a Record Mode of OS Profiler Enumeration <ul style="list-style-type: none"> ● OPF_RECMODE_STOP (0) ● OPF_RECMODE_EVENT (1) ● OPF_RECMODE_PROFILE (2) ● OPF_RECMODE_RESTART (3)

6.1.2 OpfTaskIDType

Type:	Scalar
Range	0 ~ 255
Description:	ID of Task

6.1.3 Tdd_Opf_ProfileTaskInfo

Type:	Structure
Range	-
Description:	This structure holds an information of each profiled task. OsProfiler stores worst execution time such as min ST, max jitter, max CET, max GET and max RT of each tasks. Please refer to chapter 8 for more details. ddTaskID : Task ID ddTaskPeriod : Period of Task ddMinST : Min Slack Time ddMaxJIT : Max Jitter ddMaxCET : Max Core Execution Time ddMaxGET : Max Gross Execution Time ddMaxRT : Max Response Time pTaskInfo : Pointer to a task information

6.1.4 Tdd_Opf_ProfileEventQ

Type:	Structure
Range	-
Description:	<p>This structure holds an information of each profiled event queue. OsProfiler stores event queue information of minimum Slack Time. Please refer to chapter 8.4 for more details.</p> <p>ddEventQueueSize : Size of an event queue ddLastEventPos : Position for a last event in an event queue pCurProfiledTaskInfo : Pointer to a profiled event queue ddEvent : Pointer to a profiled event queue</p>

6.2 Macro Constants

None

6.3 Functions

6.3.1 Opf_InitOsProfiler

Function Name	Opf_InitOsProfiler
Syntax	FUNC(void, OPF_CODE) Opf_InitOsProfiler(void)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This function is called to initialize OsProfiler
Preconditions	None
Configuration Dependency	None

6.3.2 Opf_StartOsProfiler

Function Name	Opf_StartOsProfiler
Syntax	FUNC(void, OPF_CODE) Opf_StartOsProfiler(void)
Service ID	0x00

Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This function is called to start OsProfiler
Preconditions	None
Configuration Dependency	None

6.3.3 Opf_StopOsProfiler

Function Name	Opf_StopOsProfiler
Syntax	FUNC(void, OPF_CODE) Opf_StopOsProfiler(void)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This function is called to stop OsProfiler
Preconditions	None
Configuration Dependency	None

6.3.4 Opf_RestartOsProfiler

Function Name	Opf_RestartOsProfiler
Syntax	FUNC(void, OPF_CODE) Opf_RestartOsProfiler(void)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	None
Description	This function is called to restart OsProfiler
Preconditions	None
Configuration Dependency	None

6.3.5 Opf_GetOsProfilerRecMode

Function Name	Opf_GetOsProfilerRecMode
Syntax	FUNC(OpfRecModeType, OPF_CODE) Opf_GetOsProfilerRecMode (void)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	None
Parameters (Inout)	None
Parameters (Out)	None
Return Value	OpfRecModeType
Description	This function is called to get OsProfiler's record mode
Preconditions	None
Configuration Dependency	None

6.3.6 Opf_GetTaskInfo


Function Name	Opf_GetTaskInfo
Syntax	FUNC(Tdd_Opf_ProfileTaskInfo *, OPF_CODE) Opf_GetTaskInfo(uint16 LddTaskID)
Service ID	0x00
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (In)	uint16 LddTaskID
Parameters (Inout)	None
Parameters (Out)	None
Return Value	Tdd_Opf_ProfileTaskInfo *
Description	This function is called to get task information
Preconditions	None
Configuration Dependency	None

6.3.7 Opf_GetProfileInfo

Function Name	Opf_GetProfileInfo
Syntax	FUNC(Tdd_Opf_ProfileEventQ *, OPF_CODE) Opf_GetProfileInfo(void)
Service ID	0x01

HYUNDAI AutoEver	Document Name : AUTOSAR Os Profiler User Manual	Page : 27/39
----------------------------	--	-----------------

<i>Sync/Async</i>	Synchronous
<i>Reentrancy</i>	Non Reentrant
<i>Parameters (In)</i>	None
<i>Parameters (Inout)</i>	None
<i>Parameters (Out)</i>	None
<i>Return Value</i>	Tdd_Opf_ProfileEventQ *
<i>Description</i>	This function is called to get profile information
<i>Preconditions</i>	None
<i>Configuration Dependency</i>	None

	Document Name : AUTOSAR Os Profiler User Manual	Page : 28/39
---	--	-----------------

7. Generator

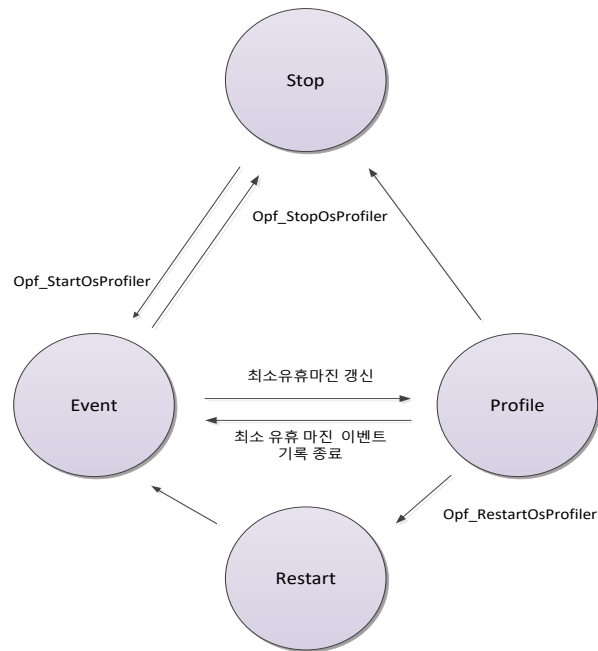
7.1 Generator Option

None

8. OS Profiler

8.1 동작 모드

OS Profiler 의 동작 모드는 하기 그림의 4가지 모드로 구성된다. OS Profiler 의 동작 모드는 API 를 이용하여, 디버거 상에서 전역 변수를 통해 확인할 수 있다.



- 전역 변수를 사용하는 경우 : Opf_GddOsProfiler.ddRecMode 저장값
- API 를 사용하는 경우 : Opf_GetOsProfilerRecMode() 의 리턴값

동작 모드로 사용되는 상수 값은 하기와 같다.

동작 모드	상수 값
OPF_RECMODE_STOP	0
OPF_RECMODE_EVENT	1
OPF_RECMODE_PROFILE	2
OPF_RECMODE_RESTART	3

8.2 System Timer Clock

OS Profiler 가 사용중인, System Timer 의 Clock 을 확인할 수 있다. 이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : Opf_GddOsProfiler.ddSTMClock 저장값 (MHz 단위)
- API 를 사용하는 경우 : Opf_GetProfileInfo() 의 리턴값인 Opf_GddOsProfiler 구조체를 통해 확인

Profiling 되는 정보들 중 일부는 System Timer Tick 값을 가지는데, System Timer 의 Clock 을 알면, 이 System Timer Tick 값을 실제 시간으로 환산할 수 있다.

8.3 이벤트 Queue

OS Profiler 가 수집하는 실시간 이벤트를 확인할 수 있다. 이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : Opf_GddOsProfiler.pEventQ
- API 를 사용하는 경우 : Opf_GetProfileInfo() 의 리턴값인 Opf_GddOsProfiler 구조체를 통해 확인

Opf_GddOsProfiler.ddEventQ 에 저장된 주요 정보는 하기와 같다.

정보	설명
Opf_GddOsProfiler.pEventQ.ddEventQueueSize	이벤트 Queue 크기
Opf_GddOsProfiler.pEventQ.ddLastEventPos	마지막 이벤트의 저장 위치
Opf_GddOsProfiler.pEventQ.ddEventCount	발생한 이벤트 수
Opf_GddOsProfiler.pEventQ.ddEvent	실제 이벤트 정보

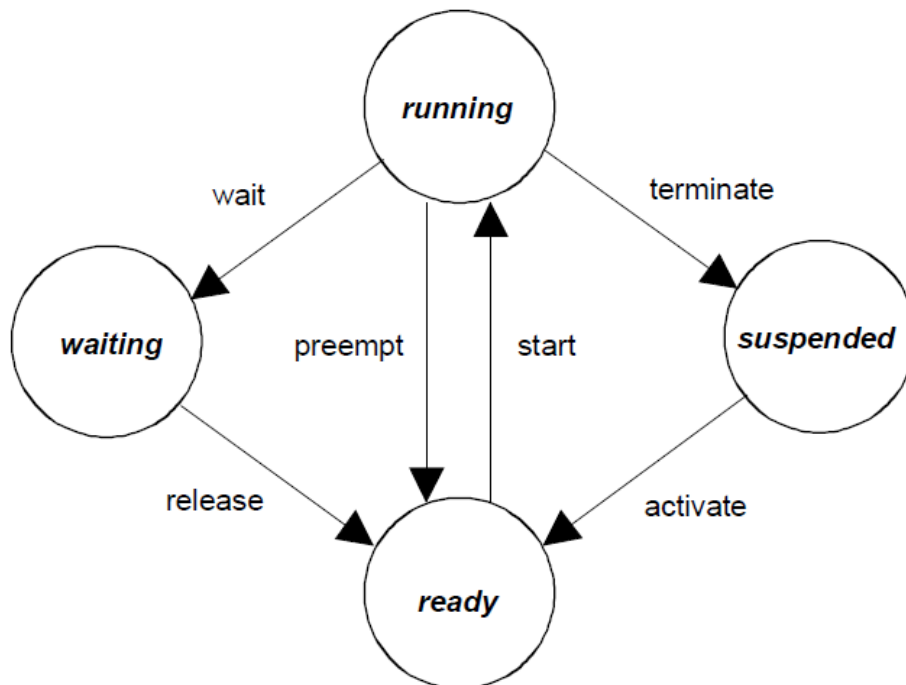
Opf_GddOsProfiler.pEventQ.ddEvent 에 저장되는 실제 이벤트 정보는 하기와 같이 System Time (이벤트 발생시의 System Time), Time Stamp (이전 이벤트와의 시간 간격), Object Type (Object 종류), Object ID (Object ID), State (Object 상태) 형식으로 저장된다.

System Time	Time Stamp	Object Type	Object ID	State
10111	111	Task	Task A	Running
10423	312	Task	Task A	Ready
10736	313	Task	Task B	Running
10850	114	ISR	ISR 1	Entry
11265	415	ISR	ISR 1	Exit
11381	116	Task	Task A	Running
12028	647	Task	Task A	Suspended
12139	111	ISR	ISR 2	Entry
12350	211	ISR	ISR 2	Exit
12470	120	Task	Task A	Running

OS Profiler 에서 이벤트 수집의 대상이 되는 Object 의 종류는 Task 와 Category2 ISR 로 제한된다.

Task 의 상태(State)는 OSEK OS / AUTOSAR OS 표준을 따라, 하기 4가지의 상태를 정의한다.

Task 상태	설명
Running	Task 가 실제 MCU Core 를 선점하고, 실제 Task 의 코드가 실행되고 있는 상태를 의미한다. 하나의 단일 코어에서 하나의 Task 만 이 상태에 있을 수 있다.
Ready	Running 상태로 천이되기를 기다리는 상태를 의미한다. 여러 개의 Task 가 동시에 Ready 상태가 될 수 있으며, 이 중, 가장 높은 우선 순위를 갖는 Task 가 Running 상태로 천이할 수 있다.
Waiting	Task 가 실행을 중지하고 특정 Event 를 기다리고 있는 상태를 의미한다. Waiting 상태에서 기다리고 있는 Event 가 발생하면, 이 Task 는 Ready 상태로 천이한다.
Suspended	Task 가 종료된 상태를 의미한다.



OS Profiler 에서는 Ready 상태를 Activated 와 Ready로 나누어 관리한다.

- Activated : Activated 되어 Running 되기를 기다리는 상태
- Ready : Running 중에 다른 Task 에 의해 선점되어 Ready 로 밀려난 상태. 추후, 권한이 있을 때 다시 Running 으로 상태 천이

Category2 ISR 의 상태는 하기 2가지 상태를 가진다.

ISR 상태	설명
Enter	Category2 ISR 실행 직전의 상태
Exit	Category2 ISR 실행 직후의 상태

8.4 Profile 이벤트 Queue

‘최소 유희 마진’ 당시의 이벤트 Queue 내용을 Profile 이벤트 Queue 에 저장한다. 이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : Opf_GddOsProfiler.pProfileEventQ
- API 를 사용하는 경우 : Opf_GetProfileInfo() 의 리턴값인 Opf_GddOsProfiler 구조체를 통해 확인

Opf_GddOsProfiler. pProfileEventQ 에 저장된 주요 정보는 하기와 같다.

정보	설명
Opf_GddOsProfiler.pProfileEventQ.ddEventQueueSize	이벤트 Queue 크기
Opf_GddOsProfiler.pProfileEventQ.ddLastEventPos	마지막 이벤트의 저장 위치
Opf_GddOsProfiler.pProfileEventQ.pCurProfiledTaskInfo	‘최소 유희 마진’ 을 발생시킨 Task 정보
Opf_GddOsProfiler.pProfileEventQ.ddEvent	‘최소 유희 마진’ 당시의 이벤트 정보

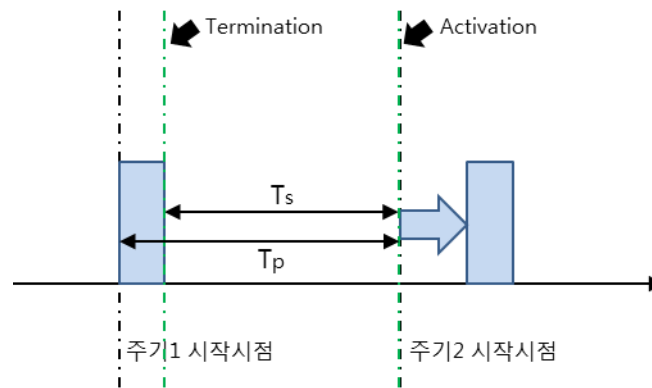
Opf_GddOsProfiler.pProfileEventQ.ddEvent 에 저장되는 ‘최소 유희 마진’ 당시의 이벤트 정보 형식은 이벤트 Queue (Opf_GddOsProfiler.pEventQ)의 형식과 동일하다.

8.5 Slack Time

주기 Task 의 ‘유희 마진’ 은 하기와 같이 계산된다. (천분율로 기록됨)

주기 Task 의 유희 마진

= ((주기 Task 의 activation 시점 - 주기 Task 의 이전 termination 시점) / Task 주기) * 1000



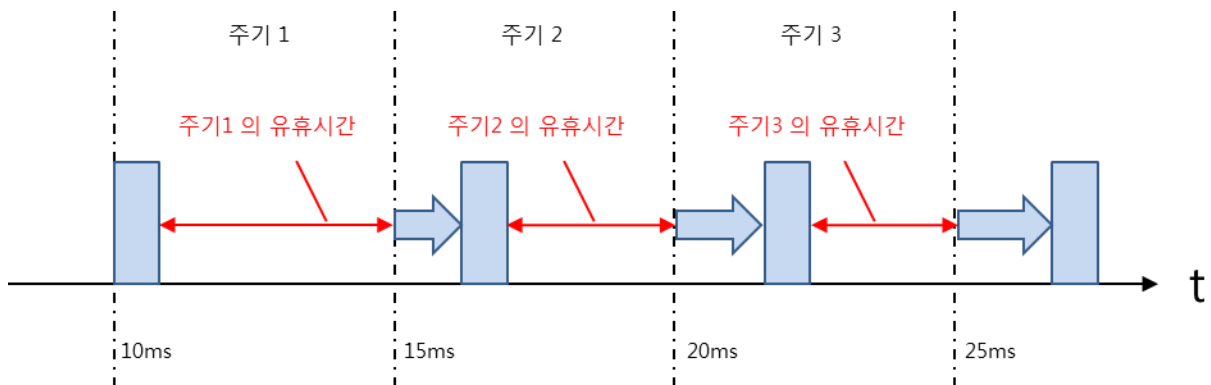
Ready

유휴 마진 = T_s / T_p 

Running

그림. Task 의 유휴 마진

예를 들면, 하기 그림에서 5ms Task 의 ‘유휴 시간’ 은 각 주기별로 각각 계산된다 (붉은색 화살표). ‘유휴 시간’ 은 주기 Task 의 activation 시점에서 주기 Task 의 이전 termination 시점을 뺀 값이고, 이 ‘유휴 시간’ 을 Task 주기로 나누어 천분율한 값이 ‘유휴 마진’ 이 된다. ‘유휴 마진’ 이 크다는 의미는 그만큼 해당 주기 Task 가 정확한 주기에 수행될 수 있다는 것을 의미한다.



Ready



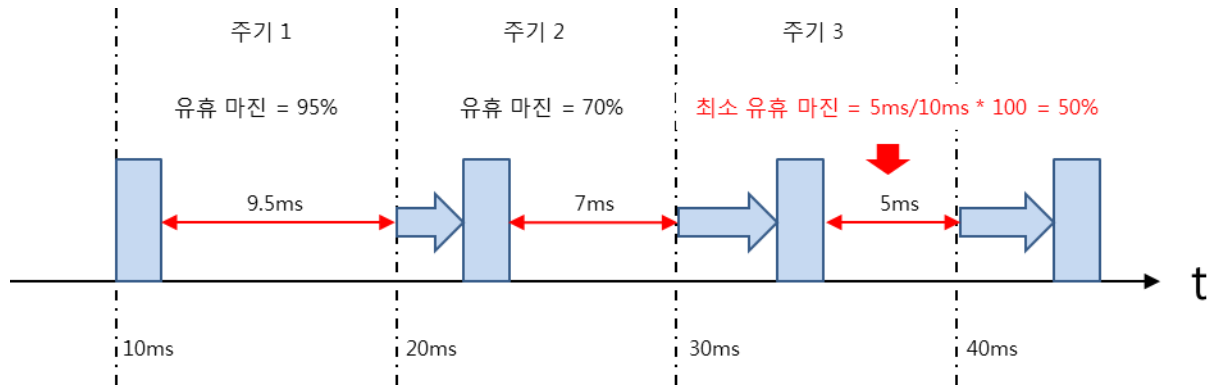
Running

그림. 주기 Task 의 유휴 마진

‘최소 유휴 마진’ 은 주기 Task 의 유휴 마진 중, 가장 작은 값을 의미한다.

주기 Task 의 최소 유휴 마진 = MIN (1st 유휴 마진, 2nd 유휴 마진, 3rd 유휴 마진, ...)

예를 들어, 하기 그림에서는 주기 1,2,3 기간 중, 주기 3의 유휴 마진이 주기 Task A의 ‘최소 유휴 마진’ 이 된다.



➡ Ready

■ Running

그림. 최소 유휴 마진

이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : Opf_GaaProfileTaskInfo[n].ddMinST (n = 0, 1, 2, ..)
- API 를 사용하는 경우 : Opf_GetTaskInfo() 의 리턴값인 Opf_GaaProfileTaskInfo 구조체를 통해 확인

8.6 Jitter

주기 Task 의 ‘Jitter’ 는 하기와 같이 계산된다. (천분율로 기록됨)

주기 Task 의 Jitter = ((Task 가 Running 되는 시점 - Task 가 Activation 되는 시점) / Task 주기) * 1000

하기 그림을 예로 들면,

주기 Task 의 Jitter = ((t2 - t1) / Tp) * 1000

이 된다.

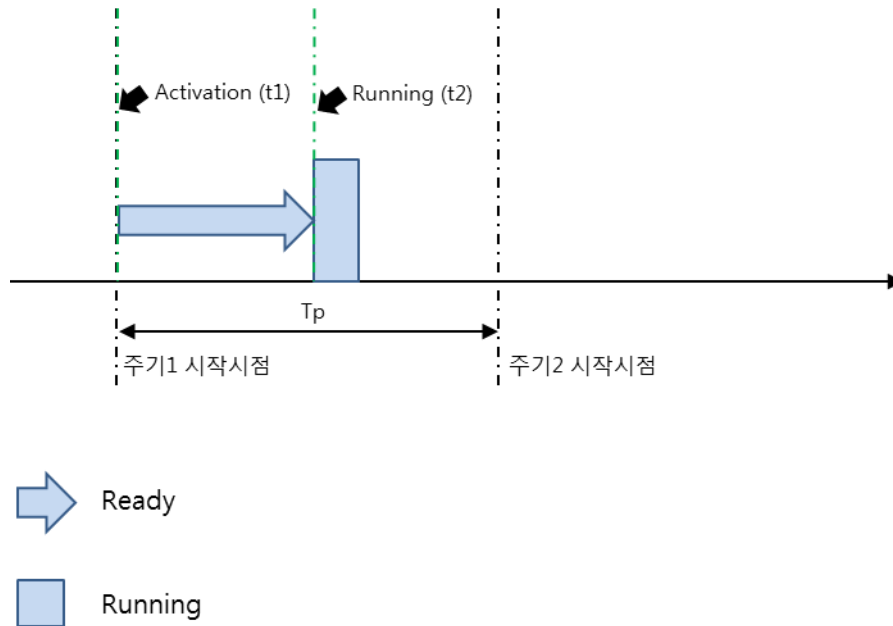


그림. 주기 Task 의 Jitter

‘최대 Jitter’ 는 주기 Task의 Jitter 중, 가장 큰 값을 의미한다.

주기 Task의 최대 Jitter = MAX (1st Jitter, 2nd Jitter, 3rd Jitter, ...)

이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : `Opf_GaaProfileTaskInfo[n].ddMaxJIT` ($n = 0, 1, 2, ..$)
- API 를 사용하는 경우 : `Opf_GetTaskInfo()` 의 리턴값인 `Opf_GaaProfileTaskInfo` 구조체를 통해 확인

8.7 CET (Core Execution Time)

‘CET (Core Execution Time)’ 은 순수하게 Task 가 수행된 시간을 의미한다. 그러므로 Task 가 수행되는 동안 다른 Task 에 의해 선점된 시간은 제외된다. 단, ISR 에 의해 선점된 시간은 제외되지 않는다.

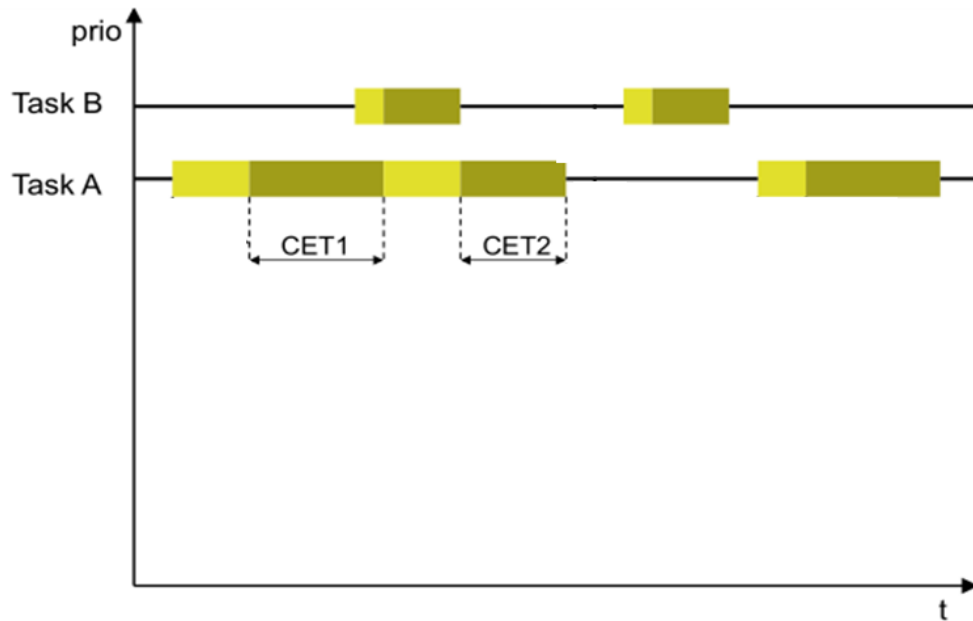


그림. CET (Core Execution Time)

‘최대 CET’ 는 주기 Task의 CET 중, 가장 큰 값을 의미한다.

주기 Task의 최대 CET = MAX (1st CET, 2nd CET, 3rd CET, ...)

이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : `Opf_GaaProfileTaskInfo[n].ddMaxCET` ($n = 0, 1, 2, \dots$)
- API 를 사용하는 경우 : `Opf_GetTaskInfo()` 의 리턴값인 `Opf_GaaProfileTaskInfo` 구조체를 통해 확인

8.8 GET (Gross Execution Time)

‘GET (Core Execution Time)’ 은 Task 가 Running 된 후, 종료(Suspended) 되기까지의 시간을 의미한다. 그러므로 Task 가 수행되는 동안 다른 Task 에 의해 선점된 시간이 포함된다.

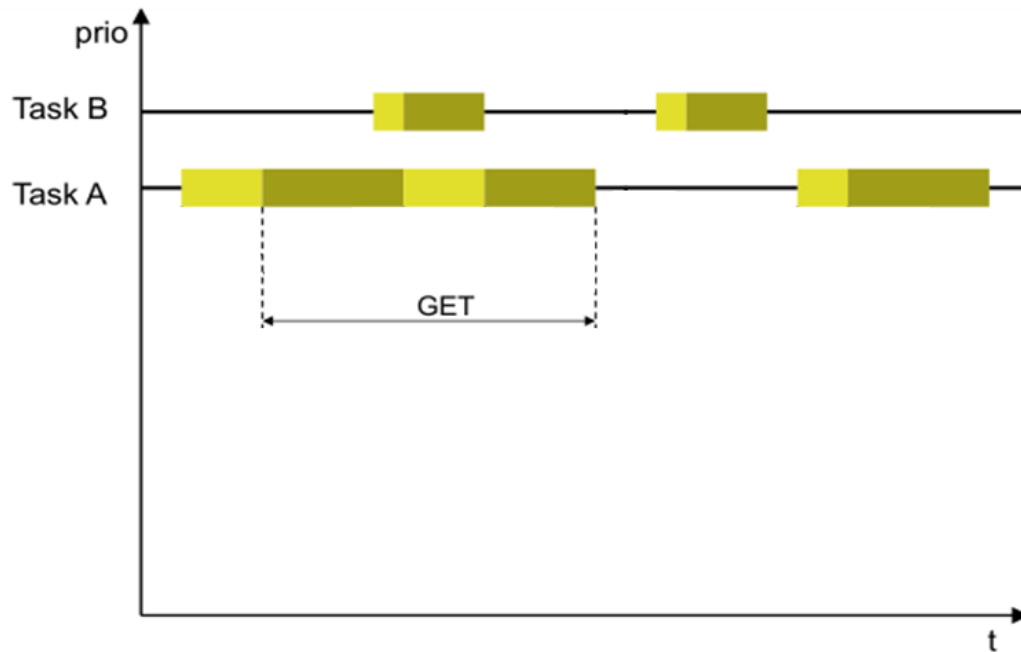


그림. GET (Gross Execution Time)

‘최대 GET’ 는 주기 Task의 GET 중, 가장 큰 값을 의미한다.

주기 Task의 최대 GET = MAX (1st GET, 2nd GET, 3rd GET, ...)

이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.

- 전역 변수를 사용하는 경우 : Opf_GaaProfileTaskInfo[n]. ddMaxGET (n = 0, 1, 2, ..)
- API 를 사용하는 경우 : Opf_GetTaskInfo() 의 리턴값인 Opf_GaaProfileTaskInfo 구조체를 통해 확인

8.9 RT (Response Time)

‘응답 시간 (Response Time)’ 은 Task 가 activation 되고, suspended 될 때까지의 시간을 의미한다. 그러므로, Task 수행 중간에 다른 Task 에 의해 선점된 시간이 모두 포함된다.

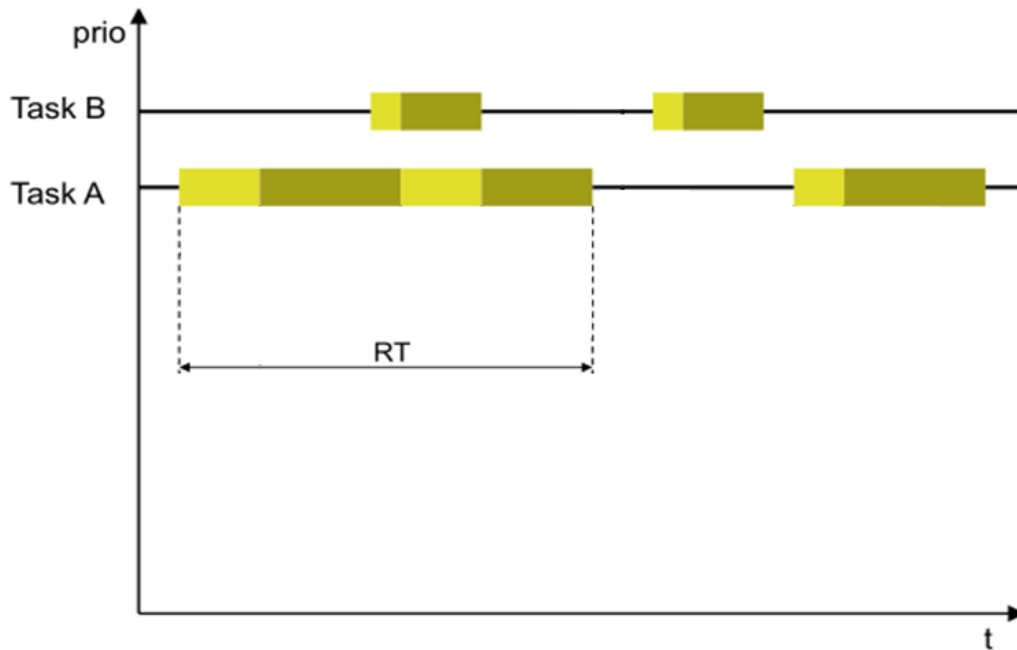


그림. 응답 시간 (Response Time)

‘최대 RT’ 는 주기 Task의 RT 중, 가장 큰 값을 의미한다.

주기 Task의 최대 RT = MAX (1st RT, 2nd RT, 3rd RT, ...)

이 정보는 API 를 이용하거나, 디버거 상에서 전역 변수를 통해 확인할 수 있다.


- 전역 변수를 사용하는 경우 : Opf_GaaProfileTaskInfo[n]. ddMaxRT (n = 0, 1, 2, ..)
- API 를 사용하는 경우 : Opf_GetTaskInfo() 의 리턴값인 Opf_GaaProfileTaskInfo 구조체를 통해 확인

8.10 Multi-Core 지원

현재 OS Profiler 기능은 Aurix Multi-Core에서의 동작을 지원한다. 단, OS Profiler 기능은 TC2xx에서 최대 3개 TC3xx에서 최대 6개 까지의 Core에 대해서 동작을 지원하며, 각 Core의 동작은 동일한 STM Frequency로 동기화가 이루어져있다고 가정한다. Multi-Core 에서의 동작은 기본적으로 Single Core에서의 동작과 동일하며 각 Core에서 발생한 이벤트를 아래와 같이 개별적으로 저장한다.

- Core0 : Opf_GddOsProfiler
- Core#n: Opf_GddOsProfilerCore#n (#n = 1 ~ 5)

주기 task는 1개의 Core에서 동작하기 때문에 Slack Time 및 RT 등의 값은 위에서 설명한 것과 동일하게 계산되며 그 값은 Opf_GaaProfileTaskInfo 를 통해 확인 할 수 있다.

	Document Name : AUTOSAR Os Profiler User Manual	Page : 39/39
---	--	-----------------

9. Appendix

9.1 설계시 유의사항

None

9.2 Exclusive Areas

None