MCAL Configuration Verification Manual for SMU

32-bit TriCore™ AURIX™ TC3xx microcontroller family

About this document

Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code of AURIX™ AUTOSAR MCAL.

Reference documents

This document should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual SMU

Table of contents

Table of contents

1 Smu driver

This chapter describes the details of the configuration data generated from the SMU driver.

1.1 File: Smu_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in 'inc' folder.

1.1.1 Macro: SMU_AR_RELEASE_MAJOR_VERSION

Table 1 SMU_AR_RELEASE_MAJOR_VERSION

Name	SMU_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the SMU implementation is based on.	
Verification method	The macro is generated with value present in ArMajorVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file with ArMajorVersion 4	`#define SMU_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: SMU_AR_RELEASE_MINOR_VERSION

Table 2 SMU_AR_RELEASE_MINOR_VERSION

Name	SMU_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the SMU implementation is based on.	
Verification method	The macro is generated with value present in ArMinorVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file with ArMinorVersion 2	`#define SMU_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: SMU_AR_RELEASE_REVISION_VERSION

Table 3 SMU_AR_RELEASE_REVISION_VERSION

Name	SMU_AR_RELEASE_REVISION_VERSION
Description	Revision version number of AUTOSAR release on which the SMU implementation is based on.

Verification method	The macro is generated with value present in ArPatchVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file with ArPatchVersion 2	`#define SMU_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: SMU_SW_MAJOR_VERSION

Table 4 SMU_SW_MAJOR_VERSION

Name	SMU_SW_MAJOR_VERSION	
Description	Major version number of the SMU module.	
Verification method	The macro is generated with value present in SwMajorVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file with SwMajorVersion 10	`#define SMU_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: SMU_SW_MINOR_VERSION

Table 5 SMU_SW_MINOR_VERSION

Name	SMU_SW_MINOR_VERSION	
Description	Minor version number of the SMU module.	
Verification method	The macro is generated with value present in SwMinorVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file with SwMinorVersion 10	`#define SMU_SW_MINOR_VERSION (10U)`

1.1.6 Macro: SMU_SW_PATCH_VERSION

Table 6 SMU_SW_PATCH_VERSION

Name	SMU_SW_PATCH_VERSION	
Description	Patch level version number of the SMU module.	
Verification method	The macro is generated with SwPatchVersion. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate Smu_Cfg.h file with SwPatchVersion 0	`#define SMU_SW_PATCH_VERSION (0U)`

1.1.7 Macro: SMU_VERSION_INFO_API

Table 7 SMU_VERSION_INFO_API

Name	SMU_VERSION_INFO_API	
Description	Enables/disables Smu_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if SmuVersionInfoApi configuration parameter is set to True else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	SmuVersionInfoApi = True	`#define SMU_VERSION_INFO_API (STD_ON)`
	SmuVersionInfoApi = False	`#define SMU_VERSION_INFO_API (STD_OFF)`

1.1.8 Macro: SMU_INIT_CHECK_API

Table 8 SMU_INIT_CHECK_API

Name	SMU_INIT_CHECK_API	
Description	Enables/disables Smu_InitCheck API	
Verification method	The macro is generated as STD_ON if SmuInitCheckApi configuration parameter is set to True else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	SmuInitCheckApi = True	`#define SMU_INIT_CHECK_API (STD_ON)`
	SmuInitCheckApi = False	`#define SMU_INIT_CHECK_API (STD_OFF)`

1.1.9 Macro: SMU_DEV_ERROR_DETECT

Table 9 SMU_DEV_ERROR_DETECT

Name	SMU_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error detection	
Verification method	The macro is generated as STD_ON if SmuDevErrorDetect configuration parameter is set to True else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	SmuDevErrorDetect = True	`#define SMU_DEV_ERROR_DETECT (STD_ON)`
	SmuDevErrorDetect = False	`#define SMU_DEV_ERROR_DETECT (STD_OFF)`

1.1.10 Macro: SMU_SAFETY_ENABLE

Table 10 SMU_SAFETY_ENABLE

Name	SMU_SAFETY_ENABLE	
Description	Enables/disables safety features	
Verification method	The macro is generated as STD_ON if SmuSafetyEnable configuration parameter is set to True else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	SmuSafetyEnable = True	`#define SMU_SAFETY_ENABLE (STD_ON)`
	SmuSafetyEnable = False	`#define SMU_SAFETY_ENABLE (STD_OFF)`

1.1.11 Macro: SMU_RUNTIME_API_MODE

Table 11 SMU_RUNTIME_API_MODE

Name	SMU_RUNTIME_API_MODE	
Description	Decides the mode of execution of Run-time APIs	
Verification method	The macro is generated as SMU_MCAL_SUPERVISOR if SmuRuntimeApiMode configuration parameter is set to SMU_MCAL_SUPERVISOR else the macro is generated as SMU_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	SmuRuntimeApiMode = SMU_MCAL_SUPERVISOR	`#define SMU_RUNTIME_API_MODE (SMU_MCAL_SUPERVISOR)`
	SmuRuntimeApiMode = SMU_MCAL_USER1	`#define SMU_RUNTIME_API_MODE (SMU_MCAL_USER1)`

1.1.12 Macro: SMU_INIT_DEINIT_API_MODE

Table 12 SMU_INIT_DEINIT_API_MODE

Name	SMU_INIT_DEINIT_API_MODE	
Description	Decides the mode of execution of Init and DeInit APIs	
Verification method	The macro is generated as SMU_MCAL_SUPERVISOR if SmuInitDeInitApiMode configuration parameter is set to SMU_MCAL_SUPERVISOR else the macro is generated as SMU_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	SmuInitDeInitApiMode = SMU_MCAL_SUPERVISOR	`#define SMU_INIT_DEINIT_API_MODE (SMU_MCAL_SUPERVISOR)`
	SmuInitDeInitApiMode = SMU_MCAL_USER1	`#define SMU_INIT_DEINIT_API_MODE (SMU_MCAL_USER1)`

1.1.13 Macro: SMU_STANDBY_SWITCH

Table 13 SMU_STANDBY_SWITCH

Name	SMU_STANDBY_SWITCH	
Description	Enables/disables the standby mode	
Verification method	The macro is generated as STD_ON if SmuStdbyEnable configuration parameter is set to True else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	SmuStdbyEnable = True	`#define SMU_STANDBY_SWITCH (STD_ON)`
	SmuStdbyEnable = False	`#define SMU_STANDBY_SWITCH (STD_OFF)`

1.1.14 Macro: SMU_CORE_FSP0_HWDIR

Table 14 SMU_CORE_FSP0_HWDIR

Name	SMU_CORE_FSP0_HWDIR	
Description	Indicates FSP0 hardware port direction for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreFSP0OutputEnable configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuCoreFSP0OutputEnable = True	`#define SMU_CORE_FSP0_HWDIR (0x1U)`
	SmuCoreFSP0OutputEnable = False	`#define SMU_CORE_FSP0_HWDIR (0x0U)`

1.1.15 Macro: SMU_CORE_FSP1_HWDIR

Table 15 SMU_CORE_FSP1_HWDIR

Name	SMU_CORE_FSP1_HWDIR	
Description	Indicates FSP1 hardware port direction for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreFSP1OutputEnable configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuCoreFSP1OutputEnable = True	`#define SMU_CORE_FSP1_HWDIR (0x1U)`
	SmuCoreFSP1OutputEnable = False	`#define SMU_CORE_FSP1_HWDIR (0x0U)`

1.1.16 Macro: SMU_CORE_FSP0_PORT_ENABLE

Table 16 SMU_CORE_FSP0_PORT_ENABLE

Name	SMU_CORE_FSP0_PORT_ENABLE	
Description	Enables/disables FSP0 hardware port for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreFSP0PortEnable configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**

	SmuCoreFSP0PortEnable = True	`#define SMU_CORE_FSP0_PORT_ENABLE (0x1U)`
	SmuCoreFSP0PortEnable = False	`#define SMU_CORE_FSP0_PORT_ENABLE (0x0U)`

1.1.17 Macro: SMU_CORE_FSP1_PORT_ENABLE

Table 17 SMU_CORE_FSP1_PORT_ENABLE

Name	SMU_CORE_FSP1_PORT_ENABLE	
Description	Enables/disables FSP1 hardware port for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreFSP1PortEnable configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuCoreFSP1PortEnable = True	`#define SMU_CORE_FSP1_PORT_ENABLE (0x1U)`
	SmuCoreFSP1PortEnable = False	`#define SMU_CORE_FSP1_PORT_ENABLE (0x0U)`

1.1.18 Macro: SMU_GLITCHFILTER_SCU

Table 18 SMU_GLITCHFILTER_SCU

Name	SMU_GLITCHFILTER_SCU	
Description	Enables/disables glitch filter through SMU_SCU for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreGlitchFilterSCU configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuCoreGlitchFilterSCU = True	`#define SMU_GLITCHFILTER_SCU (0x1U)`
	SmuCoreGlitchFilterSCU = False	`#define SMU_GLITCHFILTER_SCU (0x0U)`

1.1.19 Macro: SMU_GLITCHFILTER_SMU_STS

Table 19 SMU_GLITCHFILTER_SMU_STS

Name	SMU_GLITCHFILTER_SMU_STS	
Description	Enables/disables glitch filter through SMU_STS for SMU_core alarms	
Verification method	The macro is generated as 0x1U if SmuCoreGlitchFilterSTS configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuCoreGlitchFilterSTS = True	`#define SMU_GLITCHFILTER_SMU_STS (0x1U)`
	SmuCoreGlitchFilterSTS =	`#define SMU_GLITCHFILTER_SMU_STS`

	False	(0x0U)

1.1.20 Macro: SMU_CORE_TOTAL_ALARM_GROUPS

Table 20 SMU_CORE_TOTAL_ALARM_GROUPS

Name	SMU_CORE_TOTAL_ALARM_GROUPS	
Description	Indicates the total number of alarm groups for SMU_core domain *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as total number of alarm groups allocated to SMU_core domain based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_CORE_TOTAL_ALARM_GROUPS ((uint32)(12U))`

1.1.21 Macro: SMU_STDBY_TOTAL_ALARM_GROUPS

Table 21 SMU_STDBY_TOTAL_ALARM_GROUPS

Name	SMU_STDBY_TOTAL_ALARM_GROUPS	
Description	Indicates the total number of alarm groups for SMU_stdby domain *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as total number of alarm groups allocated to SMU_stdby domain based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_STDBY_TOTAL_ALARM_GROUPS ((uint32)(2U))`

1.1.22 Macro: SMU_CORE_TOTAL_ALARM_CONFIG_REG

Table 22 SMU_CORE_TOTAL_ALARM_CONFIG_REG

Name	SMU_CORE_TOTAL_ALARM_CONFIG_REG	
Description	Indicates the total number of alarm configuration registers for SMU_core domain *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as total number of alarm configuration registers allocated to SMU_core domain based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_CORE_TOTAL_ALARM_CONFIG_REG`

		`((uint32)(36U))`

1.1.23 Macro: SMU_STDBY_START_ALARM_GROUP

Table 23 SMU_STDBY_START_ALARM_GROUP

Name	SMU_STDBY_START_ALARM_GROUP	
Description	Indicates the first alarm group number of SMU_stdby domain *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as the first alarm group number for SMU_stdby domain based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_STDBY_START_ALARM_GROUP ((uint32)(20U))`

1.1.24 Macro: SMU_STDBY_END_ALARM_GROUP

Table 24 SMU_STDBY_END_ALARM_GROUP

Name	SMU_STDBY_END_ALARM_GROUP	
Description	Indicates the last alarm group number of SMU_stdby domain *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as the last alarm group number for SMU_stdby domain based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_STDBY_END_ALARM_GROUP ((uint32)(21U))`

1.1.25 Macro: SMU_STDBY_FSP0_OUTPUT

Table 25 SMU_STDBY_FSP0_OUTPUT

Name	SMU_STDBY_FSP0_OUTPUT	
Description	Indicates FSP0 hardware port direction for SMU_stdby alarms	
Verification method	The macro is generated as 0x1U if SmuStdbyEnableFSP0 configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuStdbyEnableFSP0 = True	`#define SMU_STDBY_FSP0_OUTPUT (0x1U)`
	SmuStdbyEnableFSP0 = False	`#define SMU_STDBY_FSP0_OUTPUT (0x0U)`

1.1.26 Macro: SMU_STDBY_FSP1_OUTPUT

Table 26 SMU_STDBY_FSP1_OUTPUT

Name	SMU_STDBY_FSP1_OUTPUT	
Description	Indicates FSP1 hardware port direction for SMU_stdby alarms	
Verification method	The macro is generated as 0x1U if SmuStdbyEnableFSP1 configuration parameter is set to True else the macro is generated as 0x0U.	
Example(s)	**Action**	**Generated output**
	SmuStdbyEnableFSP1 = True	`#define SMU_STDBY_FSP1_OUTPUT (0x1U)`
	SmuStdbyEnableFSP1 = False	`#define SMU_STDBY_FSP1_OUTPUT (0x0U)`

1.1.27 Macro: SMU_MAX_ALARM_POS

Table 27 SMU_MAX_ALARM_POS

Name	SMU_MAX_ALARM_POS	
Description	Indicates the maximum alarm position in every SMU alarm group *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as maximum alarm position allocated to every SMU alarm group based on the device.	
Example(s)	**Action**	**Generated output**
	Generate Smu_Cfg.h file	`#define SMU_MAX_ALARM_POS ((uint32)(31U))`

1.1.28 Macro: SMU_GROUP<x>_POS

Table 28 SMU_GROUP<x>_POS

Name	SMU_GROUP<x>_POS	
Description	Indicates the valid alarm positions in SMU alarm group<x> *Note: This macro is not configurable by the user.*	
Verification method	The macro is generated as a mask for valid alarm positions available for SMU alarm group<x> based on the device.	
Example(s)	**Action**	**Generated output**

	Generate Smu_Cfg.h file for TC387 device	`#define SMU_GROUP0_POS (1C07FF7U)`
		`#define SMU_GROUP1_POS (0x1c07ff7U)`
		`#define SMU_GROUP2_POS (0x1c07ff7U)`
		`#define SMU_GROUP3_POS (0x1c07ff7U)`
		`#define SMU_GROUP4_POS (0x0U)`

		#define SMU_GROUP5_POS (0x0U) #define SMU_GROUP6_POS (0x3bffdffU) #define SMU_GROUP7_POS (0xfffff007U) #define SMU_GROUP8_POS (0xfffff007U) #define SMU_GROUP9_POS (0x73802bU) #define SMU_GROUP10_POS (0x77ffffU) #define SMU_GROUP11_POS (0x37ffU) #define SMU_GROUP20_POS (0xfff0U) #define SMU_GROUP21_POS (0x1ffbfU)
	Generate Smu_Cfg.h file for TC397 device	#define SMU_GROUP0_POS (0x1c07ff7U) #define SMU_GROUP1_POS (0x1c07ff7U) #define SMU_GROUP2_POS (0x1c07ff7U) #define SMU_GROUP3_POS (0x1c07ff7U) #define SMU_GROUP4_POS (0x1c07ff7U) #define SMU_GROUP5_POS (0x1c07ff7U) #define SMU_GROUP6_POS (0x3bffdffU) #define SMU_GROUP7_POS (0xfffff1ffU) #define SMU_GROUP8_POS (0xfeffffffU) #define SMU_GROUP9_POS (0xf0f3802bU) #define SMU_GROUP10_POS (0x77ffffU) #define SMU_GROUP11_POS (0x3fffU) #define SMU_GROUP20_POS (0xfff0U) #define SMU_GROUP21_POS (0x1ffbfU)

1.1.29 Macro: SMU_ACTIVATE_RUN_STATE_FAILURE_DEM_NOTIF

Table 29 SMU_ACTIVATE_RUN_STATE_FAILURE_DEM_NOTIF

Name	SMU_ACTIVATE_RUN_STATE_FAILURE_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_ActivateRunState	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuActivateRunStateFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuActivateRunStateFailureNotification with a valid node	#define SMU_ACTIVATE_RUN_STATE_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)
	Configure SmuActivateRunStateFailureNotification with empty node	#define SMU_ACTIVATE_RUN_STATE_FAILURE_DEM_NOTIF

		(SMU_DISABLE_DEM_REPORT)

1.1.30 Macro: SMU_E_ACTIVATE_RUN_STATE_FAILURE

Table 30 SMU_E_ACTIVATE_RUN_STATE_FAILURE

Name	SMU_E_ACTIVATE_RUN_STATE_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_ActivateRunState	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuActivateRunStateFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuActivateRunStateFailureNotification = DemEventParameter_0	`#define SMU_E_ACTIVATE_RUN_STATE_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuActivateRunStateFailureNotification = DemEventParameter_2	`#define SMU_E_ACTIVATE_RUN_STATE_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.31 Macro: SMU_CLEAR_ALARM_STATUS_DEM_NOTIF

Table 31 SMU_CLEAR _ALARM_STATUS_DEM_NOTIF

Name	SMU_CLEAR_ALARM_STATUS_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_ClearAlarmStatus	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuClearAlarmStatusFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuClearAlarmStatusFailureNotification with a valid node	`#define SMU_CLEAR_ALARM_STATUS_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuClearAlarmStatusFailureNotification with empty node	`#define SMU_CLEAR_ALARM_STATUS_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.32 Macro: SMU_E_CLEAR_ALARM_STATUS_FAILURE

Table 32 SMU_E_CLEAR_ALARM_STATUS_FAILURE

Name	SMU_E_CLEAR_ALARM_STATUS_FAILURE
Description	Indicates the production error ID for failure of the API Smu_ClearAlarmStatus
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node

	value set in SmuClearAlarmStatusFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuClearAlarmStatusFailureNotification= DemEventParameter_0	`#define SMU_E_CLEAR_ALARM_STATUS_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuClearAlarmStatusFailureNotification= DemEventParameter_2	`#define SMU_E_CLEAR_ALARM_STATUS_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.33 Macro: SMU_RELEASE_FSP_DEM_NOTIF

Table 33 SMU_RELEASE_FSP_DEM_NOTIF

Name	SMU_RELEASE_FSP_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_ReleaseFSP	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuReleaseFSPFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuReleaseFSPFailureNotification with a valid node	`#define SMU_RELEASE_FSP_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuReleaseFSPFailureNotification with empty node	`#define SMU_RELEASE_FSP_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.34 Macro: SMU_E_RELEASE_FSP_FAILURE

Table 34 SMU_E_RELEASE_FSP_FAILURE

Name	SMU_E_RELEASE_FSP_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_ReleaseFSP	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuReleaseFSPFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuReleaseFSPFailureNotification = DemEventParameter_0	`#define SMU_E_RELEASE_FSP_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuReleaseFSPFailureNotification = DemEventParameter_2	`#define SMU_E_RELEASE_FSP_FAILURE`

		(DemConf_DemEventParameter_DemEventParameter_2)

1.1.35 Macro: SMU_CORE_ALIVE_FAILURE_DEM_NOTIF

Table 35 SMU_CORE_ALIVE_FAILURE_DEM_NOTIF

Name	SMU_CORE_ALIVE_FAILURE_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_CoreAliveTest	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuCoreAliveFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuCoreAliveFailureNotification with a valid node	`#define SMU_CORE_ALIVE_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuCoreAliveFailureNotification with empty node	`#define SMU_CORE_ALIVE_FAILURE_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.36 Macro: SMU_E_CORE_ALIVE_FAILURE

Table 36 SMU_E_CORE_ALIVE_FAILURE

Name	SMU_E_CORE_ALIVE_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_CoreAliveTest	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuCoreAliveFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuCoreAliveFailureNotification = DemEventParameter_0	`#define SMU_E_CORE_ALIVE_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuCoreAliveFailureNotification = DemEventParameter_2	`#define SMU_E_CORE_ALIVE_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.37 Macro: SMU_RT_STOP_FAILURE_DEM_NOTIF

Table 37 SMU_RT_STOP_FAILURE_DEM_NOTIF

Name	SMU_RT_STOP_FAILURE_DEM_NOTIF
Description	Enables/disables the Production Error reporting for failure of the API Smu_RTStop
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuRTStopFailureNotification configuration parameter else the macro is generated

	as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuRTStopFailureNotification with a valid node	`#define SMU_RT_STOP_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuRTStopFailureNotification with empty node	`#define SMU_RT_STOP_FAILURE_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.38 Macro: SMU_E_RT_STOP_FAILURE

Table 38 SMU_E_RT_STOP_FAILURE

Name	SMU_E_RT_STOP_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_RTStop	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuRTStopFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuRTStopFailureNotification = DemEventParameter_0	`#define SMU_E_RT_STOP_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuRTStopFailureNotification = DemEventParameter_2	`#define SMU_E_RT_STOP_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.39 Macro: SMU_ACTIVATE_PES_FAILURE_DEM_NOTIF

Table 39 SMU_ACTIVATE_PES_FAILURE_DEM_NOTIF

Name	SMU_ACTIVATE_PES_FAILURE_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_ActivatePES	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuActivatePESFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuActivatePESFailureNotification with a valid node	`#define SMU_ACTIVATE_PES_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuActivatePESFailureNotification with empty node	`#define SMU_ACTIVATE_PES_FAILURE_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.40 Macro: SMU_E_ACTIVATE_PES_FAILURE

Table 40 SMU_ACTIVATE_PES_FAILURE

Name	SMU_E_ACTIVATE_PES_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_ActivatePES	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuActivatePESFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuActivatePESFailureNotification = DemEventParameter_0	`#define SMU_E_ACTIVATE_PES_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuActivatePESFailureNotification = DemEventParameter_2	`#define SMU_E_ACTIVATE_PES_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.41 Macro: SMU_ACTIVATE_FSP_FAILURE_DEM_NOTIF

Table 41 SMU_ACTIVATE_FSP_FAILURE_DEM_NOTIF

Name	SMU_ACTIVATE_FSP_FAILURE_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_ActivateFSP	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuActivateFSPFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuActivateFSPFailureNotification with a valid node	`#define SMU_ACTIVATE_FSP_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuActivateFSPFailureNotification with empty node	`#define SMU_ACTIVATE_FSP_FAILURE_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.42 Macro: SMU_E_ACTIVATE_FSP_FAILURE

Table 42 SMU_E_ACTIVATE_FSP_FAILURE

Name	SMU_E_ACTIVATE_FSP_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_ActivateFSP	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuActivateFSPFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuActivateFSPFailureNotification = DemEventParameter_0	`#define SMU_E_ACTIVATE_FSP_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`

	Node in SmuActivateFSPFailureNotification = DemEventParameter_2	`#define SMU_E_ACTIVATE_FSP_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.43 Macro: SMU_SET_ALARM_STATUS_DEM_NOTIF

Table 43 SMU_SET_ALARM_STATUS_DEM_NOTIF

Name	SMU_SET_ALARM_STATUS_DEM_NOTIF	
Description	Enables/disables the Production Error reporting for failure of the API Smu_SetAlarmStatus	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuSetAlarmStatusFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuSetAlarmStatusFailureNotification with a valid node	`#define SMU_SET_ALARM_STATUS_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuSetAlarmStatusFailureNotification with empty node	`#define SMU_SET_ALARM_STATUS_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.44 Macro: SMU_E_SET_ALARM_STATUS_FAILURE

Table 44 SMU_E_SET_ALARM_STATUS_FAILURE

Name	SMU_E_SET_ALARM_STATUS_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_SetAlarmStatus	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuSetAlarmStatusFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuSetAlarmStatusFailureNotification = DemEventParameter_0	`#define SMU_E_SET_ALARM_STATUS_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuSetAlarmStatusFailureNotification = DemEventParameter_2	`#define SMU_E_SET_ALARM_STATUS_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.1.45 Macro: SMU_SFF_TEST_FAILURE_DEM_NOTIF

Table 45 SMU_RT_STOP_FAILURE_DEM_NOTIF

Name	SMU_SFF_TEST_FAILURE_DEM_NOTIF
Description	Enables/disables the Production Error reporting for failure of the API

	Smu_RegisterMonitor	
Verification method	The macro is generated as SMU_ENABLE_DEM_REPORT if a node exists in SmuSffFailureNotification configuration parameter else the macro is generated as SMU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	Configure SmuSffFailureNotification with a valid node	`#define SMU_SFF_TEST_FAILURE_DEM_NOTIF (SMU_ENABLE_DEM_REPORT)`
	Configure SmuSffFailureNotification with empty node	`#define SMU_SFF_TEST_FAILURE_DEM_NOTIF (SMU_DISABLE_DEM_REPORT)`

1.1.46 Macro: SMU_E_SFF_TEST_FAILURE

Table 46 SMU_E_RT_STOP_FAILURE

Name	SMU_E_SFF_TEST_FAILURE	
Description	Indicates the production error ID for failure of the API Smu_RegisterMonitor	
Verification method	The macro is generated as DemConf_DemEventParameter_x where x is the node value set in SmuSffFailureNotification configuration parameter.	
Example(s)	**Action**	**Generated output**
	Node in SmuSffFailureNotification = DemEventParameter_0	`#define SMU_E_SFF_TEST_FAILURE (DemConf_DemEventParameter_DemEventParameter_0)`
	Node in SmuSffFailureNotification = DemEventParameter_2	`#define SMU_E_SFF_TEST_FAILURE (DemConf_DemEventParameter_DemEventParameter_2)`

1.2 File: Smu[_<variant>]_PBcfg.c

The generated file contains all post-build configuration parameters. Post-build time configuration mechanism allows configurable functionality of SMU driver that is deployed as object code. The file is generated in ‘src’ folder.

1.2.1 Structure: Smu_Config[_<variant>]

Table 47 Smu_17_TimerIp_Config[_<variant>]

Name	Smu_Config[_<variant>]
Type	Smu_ConfigType
Description	Root configuration structure of SMU driver which will be used during initialization.
Verification method	The generated structure is present in Smu[_<variant>]_PBcfg.c file. <Variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.

Example(s)	Action	Generated output
	Configure SMU to Core0 (variant-unaware)	const Smu_ConfigType Smu_Config = { /* FSP Cfg for Smu_core*/ (uint32)0x400000U, /* AGC Cfg for SmuCore*/ (uint32)0x20000000U, /* RTC Cfg for SmuCore*/ (uint32)0x3fff02U, /* RTAC00 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC01 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC10 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC11 Cfg for SmuCore*/ (uint32)0x0U, /* CMD_STDBY config for SmuStdby*/ (uint32)0x1U, /*AlarmConfig for SmuCore*/ { 0xc03U,0xc03U,0xc03U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U,

	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U
	},
	
	/*AlarmFspConfig for SmuCore*/
	{
	0x0U, 0x0U, 0x0U, 0x0U, 0x0U,
	0x0U, 0x0U, 0x0U, 0x0U, 0x0U,
	0x0U, 0x0U },
	
	/*AlarmFspConfig for SmuStdby*/
	{
	0x160U, 0x0U }
	};

Configure SMU to Core0 (variant-aware. Variant name is ‘Petrol’)	const Smu_ConfigType Smu_Config_Petrol = { /* FSP Cfg for Smu_core*/ (uint32)0x400000U, /* AGC Cfg for SmuCore*/ (uint32)0x20000000U, /* RTC Cfg for SmuCore*/ (uint32)0x3fff02U, /* RTAC00 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC01 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC10 Cfg for SmuCore*/ (uint32)0x0U, /* RTAC11 Cfg for SmuCore*/

		(uint32)0x0U,
		
		/* CMD_STDBY config for SmuStdby*/
		(uint32)0x1U,
		
		/*AlarmConfig for SmuCore*/
		{
		0xc03U,0xc03U,0xc03U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U,
		0x0U,0x0U,0x0U
		},
		
		/*AlarmFspConfig for SmuCore*/
		{ 0x0U, 0x0U, 0x0U, 0x0U,
		0x0U, 0x0U, 0x0U, 0x0U, 0x0U,
		0x0U, 0x0U, 0x0U },
		
		/*AlarmFspConfig for SmuStdby*/
		{ 0x160U, 0x0U }
		};

1.2.1.1 Member: FSPCfg

Table 48 FSPCfg

Name	FSPCfg
Type	uint32
Description	Indicates the FSP configuration for SMU_core domain
Verification method	The structure member is generated as bitwise OR of the values corresponding to

	options selected for configuration parameters in SmuCoreFSPHandling configuration container as follows: (SmuCoreFSPPrescaler1 << 0) | (SmuCoreFSPPrescaler2 << 3) | (SmuCoreFSPSignalingMode << 5) | SmuCorePESOnFSP << 7) | (SmuCoreFSPFaultStateDuration << 22) Bits 8 to 21 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreFSPPrescaler1 = SMU_REF_CLK_FRQ_DIV_2 • SmuCoreFSPPrescaler2 = SMU_REF_CLK_FRQ_DIV_512 • SmuCoreFSPSignalingMode = SMU_FSP_BISTABLE_PROTOCOL • SmuCorePESOnFSP = SMU_FSP_PES_DISABLE • SmuCoreFSPFaultStateDuration = 0	`(uint32)0x0U`
	• SmuCoreFSPPrescaler1 = SMU_REF_CLK_FRQ_DIV_256 • SmuCoreFSPPrescaler2 = SMU_REF_CLK_FRQ_DIV_4096 • SmuCoreFSPSignalingMode = SMU_FSP_TIME_SWITCHING_PROTOCOL • SmuCorePESOnFSP = SMU_FSP_PES_ENABLE • SmuCoreFSPFaultStateDuration = 1023	`(uint32)0xffc000dfU`

1.2.1.2 Member: AGCCfg

Table 49 AGCCfg

Name	AGCCfg
Type	uint32
Description	Indicates the global configuration for SMU_core alarms
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreAlarmGlobalConfig configuration container as follows: (SmuCoreInterruptSet0 << 0) | (SmuCoreInterruptSet1 << 4) | (SmuCoreInterruptSet2 << 8) | (SmuCoreCpu0ResetRequest << 16) | (SmuCoreCpu1ResetRequest << 17) | (SmuCoreCpu2ResetRequest << 18) | (SmuCoreCpu3ResetRequest << 19) |

	(SmuCoreCpu4ResetRequest << 20) | (SmuCoreCpu5ResetRequest << 21) | (SmuCoreIGCS0ActivatePES << 24) | (SmuCoreIGCS1ActivatePES << 25) | (SmuCoreIGCS2ActivatePES << 26) | (SmuCoreNMIActivatePES << 27) | (SmuCoreCpuResetActivatePES << 28) | (SmuCoreEnableFaultToRunState << 29) Bits 3, 7, 11 to 15, 22 to 23 and 30 to 31 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreInterruptSet0 = SMU_SELECT_INT_NONE • SmuCoreInterruptSet1 = SMU_SELECT_INT_NONE • SmuCoreInterruptSet2= SMU_SELECT_INT_NONE • SmuCoreCpu0ResetRequest = False • SmuCoreCpu1ResetRequest = False • SmuCoreCpu2ResetRequest = False • SmuCoreCpu3ResetRequest = False • SmuCoreCpu4ResetRequest = False • SmuCoreCpu5ResetRequest = False • SmuCoreIGCS0ActivatePES = False • SmuCoreIGCS1ActivatePES = False • SmuCoreIGCS2ActivatePES = False • SmuCoreNMIActivatePES = False • SmuCoreCpuResetActivatePES = False • SmuCoreEnableFaultToRunState = False	`(uint32)0x0U`
	• SmuCoreInterruptSet0 = SMU_SELECT_INT0_INT1_INT2 • SmuCoreInterruptSet1 = SMU_SELECT_INT0_INT1_INT2 • SmuCoreInterruptSet2= SMU_SELECT_INT0_INT1_INT2 • SmuCoreCpu0ResetRequest = True • SmuCoreCpu1ResetRequest = True • SmuCoreCpu2ResetRequest = True • SmuCoreCpu3ResetRequest = True • SmuCoreCpu4ResetRequest = True • SmuCoreCpu5ResetRequest = True • SmuCoreIGCS0ActivatePES = True • SmuCoreIGCS1ActivatePES = True	`(uint32)0x3f3f0777U`

	• SmuCoreIGCS2ActivatePES = True • SmuCoreNMIActivatePES = True • SmuCoreCpuResetActivatePES =True • SmuCoreEnableFaultToRunState = True	

1.2.1.3 Member: RTCCfg

Table 50 RTCCfg

Name	RTCCfg	
Type	uint32	
Description	Indicates the Recovery Timer configuration for SMU_core	
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreRecoveryTimer configuration container as follows: (SmuCoreEnableRT0 << 0) | (SmuCoreEnableRT1 << 1) | (SmuCoreFSPFaultStateDuration << 8) Bits 2 to 7 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreEnableRT0 = SMU_RT_DISABLE • SmuCoreEnableRT1 = SMU_RT_DISABLE • SmuCoreFSPFaultStateDuration = 0	`(uint32)0x0U`
	• SmuCoreEnableRT0 = SMU_RT_ENABLE • SmuCoreEnableRT1 = SMU_RT_DISABLE • SmuCoreFSPFaultStateDuration = 4080	`(uint32)0xff001U`
	• SmuCoreEnableRT0 = SMU_RT_DISABLE • SmuCoreEnableRT1 = SMU_RT_ENABLE • SmuCoreFSPFaultStateDuration = 16711680	`(uint32)0xff000002U`
	• SmuCoreEnableRT0 = SMU_RT_ENABLE • SmuCoreEnableRT1 = SMU_RT_ENABLE • SmuCoreFSPFaultStateDuration = 16777215	`(uint32)0xffffff03U`

1.2.1.4 Member: RTAC00Cfg

Table 51 RTAC00Cfg

Name	RTAC00Cfg

Type	uint32	
Description	Indicates the RTAC00 configuration for SMU_core	
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreRT0Alarm configuration container as follows: (SmuCoreRT0Alarm_0/SmuCoreRT0AlarmGroupId << 0) | (SmuCoreRT0Alarm_0/SmuCoreRT0AlarmId << 4) | (SmuCoreRT0Alarm_1/SmuCoreRT0AlarmGroupId << 16) | (SmuCoreRT0Alarm_1/SmuCoreRT0AlarmId << 20) Bits 9 to 15 and 25 to 31 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreRT0Alarm_0/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT0Alarm_0/SmuCoreRT0AlarmId = 0 • SmuCoreRT0Alarm_1/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT0Alarm_1/SmuCoreRT0AlarmId = 0	`(uint32)0x0U`
	• SmuCoreRT0Alarm_0/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT0Alarm_0/SmuCoreRT0AlarmId = 10 • SmuCoreRT0Alarm_1/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT0Alarm_1/SmuCoreRT0AlarmId = 10	`(uint32)0xab00abU`

1.2.1.5 Member: RTAC01Cfg

Table 52 RTAC01Cfg

Name	RTAC01Cfg	
Type	uint32	
Description	Indicates the RTAC01 configuration for SMU_core	
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreRT0Alarm configuration container as follows: (SmuCoreRT0Alarm_2/SmuCoreRT0AlarmGroupId << 0) | (SmuCoreRT0Alarm_2/SmuCoreRT0AlarmId << 4) | (SmuCoreRT0Alarm_3/SmuCoreRT0AlarmGroupId << 16) | (SmuCoreRT0Alarm_3/SmuCoreRT0AlarmId << 20) Bits 9 to 15 and 25 to 31 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreRT0Alarm_2/SmuCoreRT0Alar	`(uint32)0x0U`

	mGroupId = SMU_ALARM_GROUP0 • SmuCoreRT0Alarm_2/SmuCoreRT0AlarmId = 0 • SmuCoreRT0Alarm_3/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT0Alarm_3/SmuCoreRT0AlarmId = 0	
	• SmuCoreRT0Alarm_2/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT0Alarm_2/SmuCoreRT0AlarmId = 10 • SmuCoreRT0Alarm_3/SmuCoreRT0AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT0Alarm_3/SmuCoreRT0AlarmId = 10	`(uint32)0xab00abU`

1.2.1.6 Member: RTAC10Cfg

Table 53 RTAC10Cfg

Name	RTAC10Cfg	
Type	uint32	
Description	Indicates the RTAC10 configuration for SMU_core	
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreRT1Alarm configuration container as follows: (SmuCoreRT1Alarm_0/SmuCoreRT1AlarmGroupId << 0) | (SmuCoreRT1Alarm_0/SmuCoreRT1AlarmId << 4) | (SmuCoreRT1Alarm_1/SmuCoreRT1AlarmGroupId << 16) | (SmuCoreRT1Alarm_1/SmuCoreRT1AlarmId << 20) Bits 9 to 15 and 25 to 31 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreRT1Alarm_0/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT1Alarm_0/SmuCoreRT1AlarmId = 0 • SmuCoreRT1Alarm_1/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT1Alarm_1/SmuCoreRT1AlarmId = 0	`(uint32)0x0U`
	• SmuCoreRT1Alarm_0/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT1Alarm_0/SmuCoreRT1AlarmId = 10 • SmuCoreRT1Alarm_1/SmuCoreRT1Alar	`(uint32)0xab00abU`

	mGroupId = SMU_ALARM_GROUP11 • SmuCoreRT1Alarm_1/SmuCoreRT1AlarmId = 10	

1.2.1.7 Member: RTAC11Cfg

Table 54 RTAC11Cfg

Name	RTAC01Cfg	
Type	uint32	
Description	Indicates the RTAC11 configuration for SMU_core	
Verification method	The structure member is generated as bitwise OR of the values corresponding to options selected for configuration parameters in SmuCoreRT1Alarm configuration container as follows: (SmuCoreRT1Alarm_2/SmuCoreRT1AlarmGroupId << 0) | (SmuCoreRT1Alarm_2/SmuCoreRT1AlarmId << 4) | (SmuCoreRT1Alarm_3/SmuCoreRT1AlarmGroupId << 16) | (SmuCoreRT1Alarm_3/SmuCoreRT1AlarmId << 20) Bits 9 to 15 and 25 to 31 are always generated as 0.	
Example(s)	**Action**	**Generated output**
	• SmuCoreRT1Alarm_2/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT1Alarm_2/SmuCoreRT1AlarmId = 0 • SmuCoreRT1Alarm_3/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP0 • SmuCoreRT1Alarm_3/SmuCoreRT1AlarmId = 0	`(uint32)0x0U`
	• SmuCoreRT1Alarm_2/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT1Alarm_2/SmuCoreRT1AlarmId = 10 • SmuCoreRT1Alarm_3/SmuCoreRT1AlarmGroupId = SMU_ALARM_GROUP11 • SmuCoreRT1Alarm_3/SmuCoreRT1AlarmId = 10	`(uint32)0xab00abU`

1.2.1.8 Member: AlarmStdbyCfg

Table 55 AlarmStandbyCfg

Name	AlarmStdbyCfg
Type	uint32
Description	Indicates the CMD_STDBY configuration for SMU_stdby domain
Verification method	The macro is generated as 0x1U if SmuStdbyEnable configuration parameter is set to True else the macro is generated as 0x0U.

Example(s)	Action	Generated output
	SmuStdbyEnable = True	`(uint32)0x1U`
	SmuStdbyEnable = False	`(uint32)0x0U`

1.2.1.9 Member: AlarmCoreConfig[SMU_CORE_TOTAL_ALARM_CONFIG_REG]

Table 56 AlarmCoreConfig[SMU_CORE_TOTAL_ALARM_CONFIG_REG]

Name	AlarmCoreConfig[SMU_CORE_TOTAL_ALARM_CONFIG_REG]	
Type	uint32	
Description	Array which stores 3-bit code for internal behavior of SMU_core alarms. *Note: Each row in the array corresponds to one of the 12 Alarm groups associated with SMU_core.* *Note: Each element in the row represents the mask for internal alarm behavior based on the number of set bits in the internal behavior code.* *Note: Each bit of element represents an alarm position.*	
Verification method	The structure member is generated as a 3*12 array such that bit<j> of every element in row<i> is based on internal alarm behavior selected in SmuCoreAlarmIntBeh configuration parameter for SmuCoreAlarmBehavior<j> of SmuCoreAlarmGroup<i>.	
Example(s)	**Action**	**Generated output**

	SmuCoreAlarmIntBeh = SMU_NA_INT_ACTION for all available alarms in all alarm groups	`{`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U,`
		`0x0U,0x0U,0x0U`
		`}`

	• SmuCoreAlarmIntBeh = SMU_IGCS1_INT_ACTION for alarm position 4 in alarm group 0
	• SmuCoreAlarmIntBeh =
		`{`
		`0x80010U,0x10U,0x80010U,`
		`0x0U,0x0U,0x0U,`

SMU_NMI_INT_ACTION for alarm position 19 in alarm group 7 • SmuCoreAlarmIntBeh = SMU_NA_INT_ACTION for all other alarms in all alarm groups	0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U, 0x0U,0x0U,0x0U }

• SmuCoreAlarmIntBeh = SMU_IGCS1_INT_ACTION for alarm position 4 in alarm group 0
• SmuCoreAlarmIntBeh = SMU_NMI_INT_ACTION for alarm position 19 in alarm group 0
• SmuCoreAlarmIntBeh = SMU_NA_INT_ACTION for all other alarms in all alarm groups
	{
	0x10U,0x10U,0x10U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x80000U,0x0U,0x80000U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U,
	0x0U,0x0U,0x0U
	}

• SmuCoreAlarmIntBeh = SMU_CPU_RESET_INT_ACTION for all available alarms in all alarm groups • SmuCoreAlarmIntBeh = SMU_NA_INT_ACTION for all unavailable alarms in all alarm groups	{ 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x3bffdffU, 0x3bffdffU, 0x3bffdffU, 0xfffff1ffU,

		0xfffff1ffU, 0xfffff1ffU, 0xfeffffffU, 0xfeffffffU, 0xfeffffffU, 0xf0f3802bU, 0xf0f3802bU, 0xf0f3802bU, 0x77ffffU, 0x77ffffU, 0x77ffffU, 0x3fffU, 0x3fffU, 0x3fffU }

1.2.1.10 Member: AlarmCoreFspConfig[SMU_CORE_TOTAL_ALARM_GROUPS]

Table 57 AlarmCoreFSPConfig[SMU_CORE_TOTAL_ALARM_GROUPS]

Name	AlarmCoreFspConfig[SMU_CORE_TOTAL_ALARM_GROUPS]	
Type	uint32	
Description	Array of FSP configurations for SMU_core alarms. *Note: Each element in the array corresponds to one of the 12 alarm groups associated with SMU_core.* *Note: Each bit of element represents an alarm position.*	
Verification method	The structure member is generated as array such that bit<j> of element<i> is based on FSP configuration selected in SmuCoreAlarmFSP configuration parameter for SmuStdbyAlarmBehavior<j> of SmuStdbyAlarmGroup<i>.	
Example(s)	**Action**	**Generated output**
	SmuCoreAlarmFSP = SMU_ALARM_FSP_DISABLED for all available alarms in all alarm groups	{ 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U, 0x0U }
	SmuCoreAlarmFSP = SMU_ALARM_FSP_ENABLED for all available alarms in all alarm groups	{ 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x1c07ff7U, 0x3bffdffU, 0xfffff1ffU, 0xfeffffffU, 0xf0f3802bU, 0x77ffffU, 0x3fffU }

1.2.1.11 Member: AlarmStdbyFspConfig[SMU_STDBY_TOTAL_ALARM_GROUPS]

Table 58 AlarmStdbyFSPConfig[SMU_STDBY_TOTAL_ALARM_GROUPS]

Name	AlarmStdbyFspConfig[SMU_STDBY_TOTAL_ALARM_GROUPS]
Type	uint32
Description	Array of FSP configurations for SMU_stdby alarms. *Note: Each element in the array corresponds to one of the 12 alarm groups*

	associated with SMU_core. *Note: Each bit of element represents an alarm position.*	
Verification method	The structure member is generated as array such that bit<j> of element<i> is based on FSP configuration selected in SmuStdbyAlarmFSP configuration parameter for SmuCoreAlarmBehavior<j> of SmuCoreAlarmGroup<i>.	
Example(s)	**Action**	**Generated output**
	SmuStdbyAlarmFSP = SMU_ALARM_FSP_DISABLED for all available alarms in all alarm groups	`{ 0x0U, 0x0U }`
	SmuStdbyAlarmFSP = SMU_ALARM_FSP_ENABLED for all available alarms in all alarm groups	`{ 0xfff0U, 0x1ffbfU }`

1.3 File: Smu[_<variant>]_PBcfg.h

The generated header file contains the declaration of the root configuration structure. Post-build time configuration mechanism allows configurable functionality of SMU driver that is deployed as object code. The file is generated in ‘inc’ folder.

1.3.1 Structure: Smu_Config[_<variant>]

Table 59 Smu_Config[_<varaint>]

Name	Smu_Config[_<variant>]	
Type	Smu_ConfigType	
Description	Declaration of root configuration structure of SMU driver which will be used during initialization.	
Verification method	The generated structure is present in Smu[_<variant>]_PBcfg.h file. The <variant> indicates the name of the post-build variant. For a variant-aware configuration the structure name is appended with the variant name. For variant-unaware configuration <variant> is ignored.	
Example(s)	**Action**	**Generated output**
	Configure SMU (variant-unaware)	`extern const Smu_ConfigType Smu_Config;`
	Configure SMU (variant-aware. Variant name is ‘Petrol’)	`extern const Smu_ConfigType Smu_Config_Petrol;`

Revision history

Major changes since the last revision

Date	Version	Description
2023-06-07	7.0	In section 1.1.30, 1.1.32, 1.1.34, 1.1.36, 1.1.38, 1.1.40, 1.1.42, 1.1.44, 1.1.46 description has been updated from DEM to production error
2020-10-27	6.0	SMU driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_CD.pdf to this document
2020-08-13	5.0	SMU_AGSTATUS_TIMEOUT parameter deleted
2019-07-09	4.0	Analyzed but no impact on Config Verification Manual
2019-02-27	1.10.0_3.0	Added PBcfg.h
2019-02-21	1.10.0_2.0	Released after fixing review comments
2019-02-20	1.10.0_1.0	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-06-07
Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
Doc_Number

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.