MCAL Configuration Verification Manual for Wdg_17_Scu

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 Wdg_17_Scu

Table of contents

1 Wdg_17_Scu driver

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

1.1 File: Wdg_17_Scu_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: WDG_17_SCU_AR_RELEASE_MAJOR_VERSION

Table 1 WDG_17_SCU_AR_RELEASE_MAJOR_VERSION

Name	WDG_17_SCU_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the Wdg_17_Scu implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file	`#define WDG_17_SCU_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: WDG_17_SCU_AR_RELEASE_MINOR_VERSION

Table 2 WDG_17_SCU_AR_RELEASE_MINOR_VERSION

Name	WDG_17_SCU_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the Wdg_17_Scu implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file	`#define WDG_17_SCU_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: WDG_17_SCU_AR_RELEASE_REVISION_VERSION

Table 3 WDG_17_SCU_AR_RELEASE_REVISION_VERSION

Name	WDG_17_SCU_AR_RELEASE_REVISION_VERSION	
Description	Revision version number of AUTOSAR release on which the Wdg_17_Scu implementation is based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file	`#define WDG_17_SCU_AR_RELEASE_REVISION_VERSION (2U)`

1.1.4 Macro: WDG_17_SCU_SW_MAJOR_VERSION

Table 4 WDG_17_SCU_SW_MAJOR_VERSION

Name	WDG_17_SCU_SW_MAJOR_VERSION	
Description	Major version number of the Wdg_17_Scu module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file with SwMajorVersion 10	`#define WDG_17_SCU_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: WDG_17_SCU_SW_MINOR_VERSION

Table 5 WDG_17_SCU_SW_MINOR_VERSION

Name	WDG_17_SCU_SW_MINOR_VERSION	
Description	Minor version number of the Wdg_17_Scu module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file with SwMinorVersion 10	`#define WDG_17_SCU_SW_MINOR_VERSION (10U)`

1.1.6 Macro: WDG_17_SCU_SW_PATCH_VERSION

Table 6 WDG_17_SCU_SW_PATCH_VERSION

Name	WDG_17_SCU_SW_PATCH_VERSION	
Description	Patch level version number of the Wdg_17_Scu module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate Wdg_17_Scu_Cfg.h file with SwPatchVersion 0	`#define WDG_17_SCU_SW_PATCH_VERSION (0U)`

1.1.7 Macro: WDG_17_SCU_SAFETY_ENABLE

Table 7 WDG_17_SCU_SAFETY_ENABLE

Name	WDG_17_SCU_SAFETY_ENABLE	
Description	Enables/disables safety features	
Verification method	The macro is generated as STD_ON if 'WdgGeneral/WdgSafetyEnable' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	WdgSafetyEnable = True	`#define WDG_17_SCU_SAFETY_ENABLE (STD_ON)`
	WdgSafetyEnable = False	`#define WDG_17_SCU_SAFETY_ENABLE (STD_OFF)`

1.1.8 Macro: WDG_17_SCU_INIT_CHECK_API

Table 8 WDG_17_SCU_INIT_CHECK_API

Name	WDG_17_SCU_INIT_CHECK_API	
Description	Enables/disables Wdg_17_Scu_InitCheck API	
Verification method	The macro is generated as STD_ON if 'WdgGeneral/WdgInitCheckApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	WdgInitCheckApi = True	`#define WDG_17_SCU_INIT_CHECK_API (STD_ON)`
	WdgInitCheckApi = False	`#define WDG_17_SCU_INIT_CHECK_API (STD_OFF)`

1.1.9 Macro: WDG_17_SCU_RUNTIME_API_MODE

Table 9 WDG_17_SCU_RUNTIME_API_MODE

Name	WDG_17_SCU_RUNTIME_API_MODE	
Description	Decides the mode of execution of Run Time API's	
Verification method	The macro is generated as WDG_17_SCU_MCAL_SUPERVISOR if 'WdgGeneral/WdgRuntimeApiMode' configuration parameter is set to 'WDG_MCAL_SUPERVISOR' else the macro is generated as WDG_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	WdgRuntimeApiMode = WDG_MCAL_SUPERVISOR	`#define WDG_17_SCU_RUNTIME_API_MODE WDG_17_SCU_MCAL_SUPERVISOR`
	WdgRuntimeApiMode = WDG_MCAL_USER1	`#define WDG_17_SCU_RUNTIME_API_MODE WDG_MCAL_USER1`

1.1.10 Macro: WDG_17_SCU_INIT_API_MODE

Table 10 WDG_17_SCU_INIT_API_MODE

Name	WDG_17_SCU_INIT_API_MODE	
Description	Decides the mode of execution of Init API.	
Verification method	The macro is generated as WDG_17_SCU_MCAL_SUPERVISOR if 'WdgGeneral/WdgInitApiMode' configuration parameter is set to 'WDG_MCAL_SUPERVISOR' else the macro is generated as WDG_MCAL_USER1.	
Example(s)	**Action**	**Generated output**
	WdgInitApiMode = WDG_MCAL_SUPERVISOR	`#define WDG_17_SCU_INIT_API_MODE WDG_MCAL_SUPERVISOR`
	WdgInitApiMode = WDG_MCAL_USER1	`#define WDG_17_SCU_INIT_API_MODE WDG_MCAL_USER1`

1.1.11 Macro: WDG_17_SCU_DEV_ERROR_DETECT

Table 11 WDG_17_SCU_DEV_ERROR_DETECT

Name	WDG_17_SCU_DEV_ERROR_DETECT	
Description	Enables/disables the Development Error Detection.	
Verification method	The macro is generated as STD_ON if 'WdgGeneral/WdgDevErrorDetect' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	WdgDevErrorDetect = True	`#define WDG_17_SCU_DEV_ERROR_DETECT (STD_ON)`
	WdgDevErrorDetect = False	`#define WDG_17_SCU_DEV_ERROR_DETECT`

		(STD_OFF)

1.1.12 Macro: WDG_17_SCU_VERSION_INFO_API

Table 12 WDG_17_SCU_VERSION_INFO_API

Name	WDG_17_SCU_VERSION_INFO_API	
Description	Enables/disables Wdg_17_Scu_GetVersionInfo API	
Verification method	The macro is generated as STD_ON if 'WdgGeneral/WdgVersionInfoApi' configuration parameter is set to 'True' else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	WdgVersionInfoApi = True	`#define WDG_17_SCU_VERSION_INFO_API (STD_ON)`
	WdgVersionInfoApi = False	`#define WDG_17_SCU_VERSION_INFO_API (STD_OFF)`

1.1.13 Macro: WDG_17_SCU_INSTANCE_ID

Table 13 WDG_17_SCU_INSTANCE_ID

Name	WDG_17_SCU_INSTANCE_ID	
Description	Instance ID of WDG module.	
Verification method	The macro is generated as a numeric value set in the configuration parameter 'WdgGeneral/WdgIndex'	
Example(s)	**Action**	**Generated output**
	Set WdgIndex as 0	`#define WDG_17_SCU_INSTANCE_ID (0U)`
	Set WdgIndex as 240	`#define WDG_17_SCU_INSTANCE_ID (240U)`

1.1.14 Macro: WDG_17_SCU_MAX_TIMERS

Table 14 WDG_17_SCU_MAX_TIMERS

Name	WDG_17_SCU_MAX_TIMERS	
Description	Maximum number of WDG timers available in hardware. *Note: This macro is not configurable by the user*	
Verification method	The macro is generated based on maximum number of WDG timers available.	
Example(s)	**Action**	**Generated output**
	Device has 4 cores, one WDG timer per core	`#define WDG_17_SCU_MAX_TIMERS (4U)`

	Device has 6 cores, one WDG timer per core	`#define WDG_17_SCU_MAX_TIMERS (6U)`

1.1.15 Macro: WDG_17_SCU_CONFIGURED_CORE<x>

Table 15 WDG_17_SCU_CONFIGURED_CORE<x>

Name	WDG_17_SCU_CONFIGURED_CORE<x>	
Description	Indicates the core on which the Wdg has been configured.	
Verification method	The macro is generated as STD_ON if CORE<x> configured else STD_OFF if CORE<x> not configured.	
Example(s)	**Action**	**Generated output**

	Configure Core 0,Core 2 and Core 3 in WdgSettingsConfig_x/ WdgCoreId	`#define WDG_17_SCU_CONFIGURED_CORE0 (STD_ON)`
		`#define WDG_17_SCU_CONFIGURED_CORE1 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE2 (STD_ON)`
		`#define WDG_17_SCU_CONFIGURED_CORE3 (STD_ON)`
		`#define WDG_17_SCU_CONFIGURED_CORE4 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE5 (STD_OFF)`

	Configure Core5 in WdgSettingsConfig_0/ WdgCoreId	`#define WDG_17_SCU_CONFIGURED_CORE0 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE1`
		`(STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE2 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE3 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE4 (STD_OFF)`
		`#define WDG_17_SCU_CONFIGURED_CORE5 (STD_ON)`

1.1.16 Macro: WDG_17_SCU_TRIG_SELECT

Table 16 WDG_17_SCU_TRIG_SELECT

Name	WDG_17_SCU_TRIG_SELECT
Description	Hardware timer selection to service WDG during window period.

Verification method	The macro is generated as WDG_17_SCU_GTM_TIMER if 'WdgGeneral/WdgTriggerTimerSelection' configuration parameter is set to 'GTM_TIMER' else the macro is generated as WDG_17_SCU_STM_TIMER.	
Example(s)	**Action**	**Generated output**
	WdgTriggerTimerSelection = GTM_TIMER	`#define WDG_17_SCU_TRIG_SELECT (WDG_17_SCU_GTM_TIMER)`
	WdgTriggerTimerSelection = STM_TIMER	`#define WDG_17_SCU_TRIG_SELECT (WDG_17_SCU_STM_TIMER)`

1.1.17 Macro: WDG_17_SCU_DISABLE_REJECT_DEM_REPORT

Table 17 WDG_17_SCU_DISABLE_REJECT_DEM_REPORT

Name	WDG_17_SCU_DISABLE_REJECT_DEM_REPORT	
Description	Enables/Disables Production error reporting for WDG_17_SCU_DISABLE_REJECT_DEM_REPORT.	
Verification method	The macro is generated as WDG_17_SCU_ENABLE_DEM_REPORT if node exists in 'WdgDemEventParameterRefs/WdgDemEventParameterRefs_0/ WDG_E_DISABLE_REJECTED' else the macro is generated as WDG_17_SCU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	In WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0 configure DEM for WDG_E_DISABLE_REJECTED	`#define WDG_17_SCU_DISABLE_REJECT_DEM_REPORT (WDG_17_SCU_ENABLE_DEM_REPORT)`
	In WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0 if DEM not configured.	`#define WDG_17_SCU_DISABLE_REJECT_DEM_REPORT (WDG_17_SCU_DISABLE_DEM_REPORT)`

1.1.18 Macro: WDG_17_SCU_E_DISABLE_REJECTED

Table 18 WDG_17_SCU_E_DISABLE_REJECTED

Name	WDG_17_SCU_E_DISABLE_REJECTED	
Description	Specifies the value configured for Production error for watchdog disable failure.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'WdgDemEventParameterRefs_0/ WDG_E_DISABLE_REJECTED'.	
Example(s)	**Action**	**Generated output**
	Configure node in WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0/	`#define WDG_17_SCU_E_DISABLE_REJECTED (DemConf_DemEventParameter_`

	WDG_E_DISABLE_REJECTED = DemEventParameter_0	DemEventParameter_0)
	Configure node in WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0/ WDG_E_DISABLE_REJECTED = DemEventParameter_1	#define WDG_17_SCU_E_DISABLE_REJECTED (DemConf_DemEventParameter_ DemEventParameter_1)

1.1.19 Macro: WDG_17_SCU_MODE_FAIL_DEM_REPORT

Table 19 WDG_17_SCU_MODE_FAIL_DEM_REPORT

Name	WDG_17_SCU_MODE_FAIL_DEM_REPORT	
Description	Enables/Disables Production error reporting for WDG_17_SCU_MODE_FAIL_DEM_REPORT.	
Verification method	The macro is generated as WDG_17_SCU_ENABLE_DEM_REPORT if node exists in 'WdgDemEventParameterRefs/WdgDemEventParameterRefs_0/WDG_E_MODE_FAILED' else the macro is generated as WDG_17_SCU_DISABLE_DEM_REPORT.	
Example(s)	**Action**	**Generated output**
	In WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0 configure DEM for WDG_E_MODE_FAILED	#define WDG_17_SCU_MODE_FAIL_DEM_REPORT (WDG_17_SCU_ENABLE_DEM_REPORT)
	In WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0 if DEM not configured.	#define WDG_17_SCU_MODE_FAIL_DEM_REPORT (WDG_17_SCU_DISABLE_DEM_REPORT)

1.1.20 Macro: WDG_17_SCU_E_MODE_FAILED

Table 20 WDG_17_SCU_E_MODE_FAILED

Name	WDG_17_SCU_E_MODE_FAILED	
Description	Specifies the value configured for Production error for watchdog mode failure.	
Verification method	The macro is generated as 'DemConf_DemEventParameter_x' where x is the node value configured in 'WdgDemEventParameterRefs/WdgDemEventParameterRefs_0/ WDG_E_MODE_FAILED'.	
Example(s)	**Action**	**Generated output**
	Configure node in WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0/ WDG_E_DISABLE_REJECTED = DemEventParameter_0	#define WDG_17_SCU_E_MODE_FAILED (DemConf_DemEventParameter_ DemEventParameter_0)
	Configure node in	#define WDG_17_SCU_E_MODE_FAILED

	WdgDemEventParameterRefs/ WdgDemEventParameterRefs_0/ WDG_E_DISABLE_REJECTED = DemEventParameter_1	`(DemConf_DemEventParameter_ DemEventParameter_1)`

1.2 File: Wdg_17_Scu[_<variant>]_PBcfg.c

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

1.2.1 Structure: Wdg_17_Scu_Config_x[_<variant>]

Table 21 Wdg_17_Scu_Config_x[_<variant>]

Name	Wdg_17_Scu_Config_x[_<variant>]	
Type	Wdg_17_Scu_ConfigType	
Description	Root configuration structure of WDG driver which will be used during initialization.	
Verification method	The generated structure is present in Wdg_17_Scu[_<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
	WdgTriggerTimerSelection as GTM_TIMER
	In WdgSettingsConfig container configure
	WdgCoreId = 0
	WdgCPUDisableAllowed = True
	WdgCPUInitialTimeout=2.0
	WdgCPUMaxTimeout=20.0
	WdgCPUInitialPassowrd=111
	WdgDefaultMode = WDGIF_SLOW_MODE
	WdgFastModeTimeoutValue =0.16
	WdgSlowModeTimeoutValue = 1.0
	WdgSlowRefreshTime = 0.33
	WdgFastRefreshTime = 0.11
	(variant-unaware)
		`const Wdg_17_Scu_ConfigType Wdg_17_Scu_Config_0 =`
		`{`
		`Wdg_GtmConfig_0,`
		`/*FastMode reload value*/`
		`(uint16)3036,`
		`/*SlowMode reload value*/`
		`59432,`
		`/*Fast refresh time*/`
		`110,`
		`/*Slow refresh time*/`
		`330,`
		`/*Wdg initial timeout*/`
		`2000,`
		`/*Wdg maximum timeout*/`
		`20000,`
		`/*Default mode*/`
		`WDGIF_SLOW_MODE,`
		`/*Core Disable allowed status*/`
		`TRUE,`
		`/*Core Id*/`
		`0,`

		/*CPU Wdg Password*/ 111 };

	Configure WdgTriggerTimerSelection as STM_TIMER
	In WdgSettingsConfig container configure
	WdgCoreId = 0
	WdgCPUDisableAllowed = True
	WdgCPUInitialTimeout=5.0
	WdgCPUMaxTimeout=65.0
	WdgCPUInitialPassowrd=111
	WdgDefaultMode = WDGIF_SLOW_MODE
	WdgFastModeTimeoutValue =0.16
	WdgSlowModeTimeoutValue = 2.0
	WdgSlowRefreshTime = 1.0
	WdgFastRefreshTime = 0.15
	(variant-aware. Variant name is 'Petrol')
		const struct Wdg_17_Scu_ConfigType Wdg_17_Scu_Config_0_Petrol =
		{
		/*STM compare Reg used for Servicing*/
		&Wdg_StmConfig_0_Petrol,
		
		/*FastMode reload value*/
		(uint16)3036,
		/*SlowMode reload value*/
		53328,
		/*Fast refresh time*/
		150,
		/*Slow refresh time*/
		1000,
		/*Wdg initial timeout*/
		5000,
		/*Wdg maximum timeout*/
		65000,
		/*Default mode*/
		WDGIF_SLOW_MODE,
		/*Core Disable allowed status*/
		TRUE,
		/*Core Id*/
		0,
		/*CPU Wdg Password*/
		111
		};

1.2.1.1 Member: WdgStmConfig_x[_<variant>]

Table 22 WdgStmConfig_x[_<variant>]

Name	WdgStmConfig_x[_<variant>]
Type	Mcu_17_Stm_TimerConfigType *
Description	Pointer to STM timer configuration structure.

Verification method	The generated structure member is present in the Wdg_17_Scu_Config_x[_<variant>] structure. The structure member is generated as a pointer to STM timer configuration.	
Example(s)	**Action**	**Generated output**
	Configure WdgTriggerTimerSelection = STM_TIMER for Core 0 (variant-unaware)	`{ /*STM compare Reg used for Servicing*/ &Wdg_StmConfig_0, }`
	Configure WdgTriggerTimerSelection = STM_TIMER for Core 0 (variant-aware. Variant name is 'Petrol')	`{ /*STM compare Reg used for Servicing*/ &Wdg_StmConfig_0_Petrol, }`

1.2.1.2 Member: Wdg_GtmConfig_x[_<variant>]

Table 23 Wdg_GtmConfig_x[_<variant>]

Name	Wdg_GtmConfig_x[_<variant>]	
Type	Mcu_17_Gtm_TomAtomChConfigType *	
Description	Pointer to GTM timer configuration structure.	
Verification method	The generated structure member is present in the Wdg_17_Scu_Config_x[_<variant>] structure. The structure member is generated as a pointer to GTM timer configuration.	
Example(s)	**Action**	**Generated output**
	Configure WdgTriggerTimerSelection = GTM_TIMER for Core 0 (variant-unaware)	`{ Wdg_GtmConfig_0, }`
	Configure WdgTriggerTimerSelection = GTM_TIMER for Core 0 (variant-aware. Variant name is 'Petrol')	`{ Wdg_GtmConfig_0_Petrol, }`

1.2.1.3 Member: FastModeReloadValue

Table 24 FastModeReloadValue

Name	FastModeReloadValue
Type	uint16
Description	Fast mode reload value in ticks.

Verification method	The structure member is generated as the reload value of WDG timer in fast mode. Steps to calculate FastModeReloadValue 1. TimeoutVal = WdgFastModeTimeoutValue * 1000. 2. Reload value is calculated as follows ReloadValue = (((System Clock * TimeoutVal)/1000)/Clock divider) where System Clock is frequency of SPB in MHz and Clock divider is 256 for fast mode. 3. ReloadValue =Ceiling(ReloadValue). 4. FastModeReloadValue = 65536 – (Reload Value).	
Example(s)	**Action**	**Generated output**

	Configure WdgFastModeTimeoutValue = 0.05 in WdgSettingsConfig_0/ WdgSettingsFast container	`{`
		`/*FastMode reload value*/`
		`(uint16)46004,`
		`}`

1.2.1.4 Member: SlowModeReloadValue

Table 25 SlowModeReloadValue

Name	SlowModeReloadValue	
Type	uint16	
Description	Slow mode reload value in ticks.	
User configurable	Yes	
Verification method	The structure member is generated as the reload value of WDT in slow mode. Steps to calculate SlowModeReloadValue 1. TimeoutVal = WdgSlowModeTimeoutValue * 1000. 2. Reload value is calculated as follows ReloadValue = (((System Clock * TimeoutVal)/1000)/Clock divider) where System Clock is frequency of SPB in MHz and Clock divider is 16384 for slow mode. 3. ReloadValue =Ceiling(ReloadValue). FastModeReloadValue = 65536 – (Reload Value).	
Example(s)	**Action**	**Generated output**

	Configure WdgSlowModeTimeoutValue = 1.0 in WdgSettingsConfig_0/ WdgSettingsSlow container	`{`
		`/*SlowMode reload value*/`
		`59432,`
		`}`

1.2.1.5 Member: FastModeRefreshTime

Table 26 FastModeRefreshTime

Name	FastModeRefreshTime	
Type	uint16	
Description	Fast mode GTM/STM callback period in seconds for WDTx.	
Verification method	The structure member is generated as value in terms of GTM/STM callback period. • WdgFastRefreshTime = (WdgFastRefreshTime * 1000).	
Example(s)	**Action**	**Generated output**

	Configure WdgFastRefreshTime = 0.001 in WdgSettingsConfig_0/ WdgTriggerTimerSetting container	`{`
		`/*Fast refresh time*/`
		`1,`
		`}`

1.2.1.6 Member: SlowModeRefreshTime

Table 27 SlowModeRefreshTime

Name	SlowModeRefreshTime	
Type	uint16	
Description	Slow mode GTM/STM callback period in seconds for WDTx.	
Verification method	The structure member is generated as value in terms of GTM/STM callback period. • WdgSlowRefreshTime = (WdgSlowRefreshTime * 1000).	
Example(s)	**Action**	**Generated output**

	Configure WdgSlowRefreshTime = 0.2 in WdgSettingsConfig_0/ WdgTriggerTimerSetting container	`{`
		`/*Slow refresh time*/`
		`200,`
		`}`

1.2.1.7 Member: InitialRefreshTime

Table 28 InitialRefreshTime

Name	InitialRefreshTime	
Type	uint16	
Description	This is the initial window period that is active as soon as Wdg_17_Scu_Init is called for the core. It is used to calculate the value of the trigger counter which is used to service the WDT just after initialization.	
Verification method	The structure member is generated as value in terms of initial window period used to calculate the trigger counter. • WdgCPUInitialTimeout = (WdgCPUInitialTimeout * 1000).	
Example(s)	**Action**	**Generated output**
	Configure WdgCPUInitialTimeout = 2.0 in WdgSettingsConfig_0 container	`{` `/*Wdg initial timeout*/`

		2000, }

1.2.1.8 Member: MaxTimeOutTime

Table 29 MaxTimeOutTime

Name	MaxTimeOutTime	
Type	uint16	
Description	This is the maximum window period for the core specific watchdog timer.	
Verification method	The structure member is generated as maximum window period. • WdgCPUMaxTimeout = (WdgCPUMaxTimeout * 1000).	
Example(s)	**Action**	**Generated output**

	Configure WdgCPUMaxTimeout = 20.0 in WdgSettingsConfig_0 container	{
		/* Wdg maximum timeout */
		20000,
		}

1.2.1.9 Member: DefaultMode

Table 30 DefaultMode

Name	DefaultMode	
Type	WdgIf_ModeType	
Description	Default mode of WDG Driver initialization for WDG timer.	
Verification method	The structure member is generated as default mode for WDG timer.	
Example(s)	**Action**	**Generated output**

	Configure WdgDefaultMode = WDGIF_SLOW_MODE in WdgSettingsConfig_0 container	{
		/*Default mode*/
		WDGIF_SLOW_MODE,
		}

1.2.1.10 Member: WdgDisableAllowed

Table 31 WdgDisableAllowed

Name	WdgDisableAllowed	
Type	boolean	
Description	Enables/disables the permission to disable the watchdog.	
Verification method	The macro is generated as TRUE if WdgDisableAllowed configuration parameter is set to ‘True’ else the macro is generated as FALSE.	
Example(s)	**Action**	**Generated output**
	Configure	{

	WdgCPUDisableAllowed = TRUE in WdgSettingsConfig_0 container	/*Core Disable allowed status*/ TRUE, }

	Configure WdgCPUDisableAllowed = FALSE in WdgSettingsConfig_0 container	{
		/*Core Disable allowed status*/
		FALSE,
		}

1.2.1.11 Member: WdgCoreID

Table 32 WdgCoreID

Name	WdgCoreID	
Type	uint32	
Description	CoreID of the WDG timer.	
Verification method	The macro is generated as a numeric value set in the configuration parameter 'WdgSettingsConfig_0/ WdgCoreId'	
Example(s)	**Action**	**Generated output**

	Configure WdgCoreId = 4 in WdgSettingsConfig_0 container	{
		/*Core Id*/
		4,
		}

1.2.1.12 Member: WdgPassword

Table 33 WdgPassword

Name	WdgPassword	
Type	uint32	
Description	Initial password for the password access of the CPU WDG.	
Verification method	The macro is generated as a numeric value set in the configuration parameter 'WdgSettingsConfig_0/ WdgCPUInitialPassowrd'	
Example(s)	**Action**	**Generated output**

	Configure WdgCPUInitialPassword= 111 in WdgSettingsConfig_0 container	{
		/*CPU Wdg Password*/
		111
		}

1.2.2 Structure: Wdg_GtmConfig_ <x>[_<variant>] [2]

Table 34 Wdg_GtmConfig_ <x>[_<variant>] [2]

Name	Wdg_GtmConfig_ <x>[_<variant>] [2]	
Type	Mcu_17_Gtm_TomAtomChConfigType	
Description	Contains GTM timer configuration information for slow mode and fast mode.	
Verification method	The generated file has this structure if GTM timers is selected as WdgTriggerTimerSelection assigned to Core <x>. <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 WdgTriggerTimerSelection = GTM_TIMER for Core 0
	In GtmTimerConfiguration_0/ GtmTimerOutputModule Configuration
	Configure GtmTimerUsed=TOM0Channel0
	GtmTimerClockSelect= GTM_FIXED_CLOCK_2
	(variant-unaware)
		`static const Mcu_17_Gtm_TomAtomChConfigType Wdg_GtmConfig_0[2] =`
		`{`
		`  /*GTM channel structure for Slow*/`
		`  {`
		`  /*Gtm module used to services wdg*/`
		`  MCU_GTM_TIMER_TOM,`
		`  /* Timer Number Module No | Timer Channel No */`
		`  0x0,`
		`  /* Ctrl register load value */`
		`  10240,`
		`  /*Timer Channel CN0 value*/`
		`  0x0U,`
		`  /*Timer Channel CM0 value*/`
		`  64453U,`
		`  /*Timer Channel CM1 value*/`
		`  0x0U,`
		`  /*Timer Channel SR0 value*/`
		`  64453U,`
		`  /*Timer Channel SR1 value*/`
		`  0x0U,`
		`  /*Channel to Port Value*/`
		`  0x0U,`
		`  /*Timer Channel Interrupt Enable value*/`
		`  0x81U`
		`  },`
		`  /*GTM channel structure for Fast*/`
		`  {`
		`  /*Gtm module used to services wdg*/`

	MCU_GTM_TIMER_TOM,
	/* Timer Number Module No | Timer Channel No */
	0x0,
	/* Ctrl register load value */
	10240,
	/*Timer Channel CN0 value*/
	0x0U,
	/*Timer Channel CM0 value*/
	21484U,
	/*Timer Channel CM1 value*/
	0x0U,
	/*Timer Channel SR0 value*/
	21484U,
	/*Timer Channel SR1 value*/
	0x0U,
	/*Channel to Port Value*/
	0x0U,
	/*Timer Channel Interrupt Enable value*/
	0x81U
	}
	};

Configure WdgTriggerTimerSelection = GTM_TIMER for Core 0 In GtmTimerConfiguration_0/ GtmTimerOutputModule Configuration Configure GtmTimerUsed=TOM0Channel0 GtmTimerClockSelect= GTM_FIXED_CLOCK_2 (variant-aware. Variant name is 'Petrol')	static const Mcu_17_Gtm_TomAtomChConfigType Wdg_GtmConfig_0_Petrol[2] = { /*GTM channel structure for Slow*/ { /*Gtm module used to services wdg*/ MCU_GTM_TIMER_TOM, /* Timer Number Module No | Timer Channel No */ 0x0, /* Ctrl register load value */ 10240, /*Timer Channel CN0 value*/ 0x0U, /*Timer Channel CM0 value*/

```
    64453U,
    /*Timer Channel CM1 value*/
    0x0U,
    /*Timer Channel SR0 value*/
    64453U,
    /*Timer Channel SR1 value*/
    0x0U,
    /*Channel to Port Value*/
    0x0U,
    /*Timer Channel Interrupt Enable
value*/
    0x81U
  },
  /*GTM channel structure for Fast*/
  {
    /*Gtm module used to services wdg*/
    MCU_GTM_TIMER_TOM,
    /* Timer Number Module No | Timer
Channel No */
    0x0,
    /* Ctrl register load value */
    10240,
    /*Timer Channel CN0 value*/
    0x0U,
    /*Timer Channel CM0 value*/
    21484U,
    /*Timer Channel CM1 value*/
    0x0U,
    /*Timer Channel SR0 value*/
    21484U,
    /*Timer Channel SR1 value*/
    0x0U,
     /*Channel to Port Value*/
    0x0U,
    /*Timer Channel Interrupt Enable
value*/
    0x81U
  }
};
```

1.2.2.1 Member: TimerType

Table 35 TimerType

Name	TimerType	
Type	Mcu_17_Gtm_TimerOutType	
Description	TOM/ATOM channel used to service the watchdog.	
Verification method	The structure member is generated with TOM/ATOM timer type used to service the watchdog.	
Example(s)	**Action**	**Generated output**
	Configure GtmTimerUsed = McuGtmTomAllocationConf_0 /McuGtmTomChannelAllocation Conf_0 in GtmTimerConfiguration_0	`static const Mcu_17_Gtm_TomAtomChConfigType Wdg_GtmConfig_0[2] = { /*GTM channel structure for Slow*/ { /*Gtm module used to services wdg*/ MCU_GTM_TIMER_TOM, } /*GTM channel structure for Fast*/ { /*Gtm module used to services wdg*/ MCU_GTM_TIMER_TOM, } }`
	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0/ McuGtmAtomChannelAllocation Conf_0 in GtmTimerConfiguration_0	`static const Mcu_17_Gtm_TomAtomChConfigType Wdg_GtmConfig_0[2] = { /*GTM channel structure for Slow*/ { /*Gtm module used to services wdg*/ MCU_GTM_TIMER_ATOM, } /*GTM channel structure for Fast*/ {`

		/*Gtm module used to services wdg*/ MCU_GTM_TIMER_ATOM, } }

1.2.2.2 Member: TimerId

Table 36 TimerId

Name	TimerId	
Type	Mcu_17_Gtm_TimerChIdentifierType	
Description	TOM/ATOM channel identifier.	
Verification method	The structure member is generated as numeric value used to represent timer module number and channel number.	
Example(s)	**Action**	**Generated output**

	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0 /McuGtmAtomChannelAllocationConf_4 in GtmTimerConfiguration_0	/*GTM channel structure for Slow*/
		{
		/* Timer Number Module No | Timer Channel No */
		0x4,
		}
		/*GTM channel structure for Fast*/
		{
		/* Timer Number Module No | Timer Channel No */
		0x4,
		}

	Configure GtmTimerUsed = McuGtmTomAllocationConf_1 /McuGtmTomChannelAllocationConf_6 in GtmTimerConfiguration_0	/*GTM channel structure for Slow*/
		{
		/* Timer Number Module No | Timer Channel No */
		0x106,
		}
		/*GTM channel structure for Fast*/
		{
		/* Timer Number Module No | Timer Channel No */
		0x106,,

		}

1.2.2.3 Member: TimerChCtrlReg

Table 37 TimerChCtrlReg

Name	TimerChCtrlReg	
Type	uint32	
Description	TOM/ATOM channel control registers value.	
Verification method	The structure member is generated as value of the control register for TOM/ATOM channel. Steps to calculate TimerChCtrlReg • Fixed value for TimerChCtrlReg is 0x00000802 for ATOM and 0x00000800 for TOM • Based on the GtmTimerClockSelect, value of clock select is left shifted by 12 and OR'ed with TImerChCtrlReg. TImerChCtrlReg = (TImerChCtrlReg | (ClockSelect<<12))	
Example(s)	**Action**	**Generated output**

	Configure GtmTimerUsed = McuGtmTomAllocationConf_1 /McuGtmTomChannelAllocation Conf_6 and GtmTimerClockSelect = GTM_FIXED_CLOCK_2 in GtmTimerConfiguration_0	/*GTM channel structure for Slow*/
		{
		/* Ctrl register load value */
		10240,
		}
		/*GTM channel structure for Fast*/
		{
		/* Ctrl register load value */
		10240,
		}

	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0 / McuGtmAtomChannelAllocatio nConf_0 and GtmTimerClockSelect = GTM_CONFIGURABLE_CLOCK_4 in GtmTimerConfiguration_0	/*GTM channel structure for Slow*/
		{
		/* Ctrl register load value */
		18434,
		}
		/*GTM channel structure for Fast*/
		{
		/* Ctrl register load value */
		18434,
		}

1.2.2.4 Member: TimerChCN0Reg

Table 38 TimerChCN0Reg

Name	TimerChCN0Reg	
Type	uint32	
Description	TOM/ATOM channel CN0 register value.	
Verification method	The structure member is generated as value of the CN0 register for TOM/ATOM channel. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel CN0 value*/`
		`0x0U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Timer Channel CN0 value*/`
		`0x0U,`
		`}`

1.2.2.5 Member: TimerChCM0Reg

Table 39 TimerChCM0Reg

Name	TimerChCM0Reg
Type	uint32
Description	TOM/ATOM channel CM0 register value.
Verification method	The structure member is generated as value of the CM0 register for TOM/ATOM channel. Steps to calculate TimerChCM0Reg • GTM frequency calculation fGtm=((McuGTMFrequency * GtmDenominator)/ GtmNumerator) fGtm=(fGtm / GtmClusterDivVal) fGtm= fGtm/ ClockDivider • TimerChCM0Reg value is calculated based on RefreshTime. Refresh time can be SlowRefreshTime/ FastRefreshTime depending on the mode of the watchdog. TimerChCM0Reg = ((RefreshTime*1000) * fGtm)/1000)

Example(s)	**Action**	**Generated output**

	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0 / McuGtmAtomChannelAllocation Conf_0 and GtmTimerClockSelect = GTM_CONFIGURABLE_CLOCK_4 in GtmTimerConfiguration_0
	GTM frequency = 50MHZ
	
	In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITH_ DIV_SEL2
	
	Configure GtmClusterConfClock4Src= CMU_CONF_CLOCK4_SEL0 in GtmGlobalConfiguration/*[1]/GtmClusterConf/ GtmClusterConf_1/ GtmClusterConfClockSetting
	
	In WdgSettingsConfig_0/WdgTriggerTimerSetting
	Configure WdgSlowRefreshTime=0.33 WdgFastRefreshTime=0.11
		`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel CM0 value*/`
		`16500000U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Timer Channel CM0 value*/`
		`5500000U,`
		`}`

	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0 / McuGtmAtomChannelAllocation Conf_0 and GtmTimerClockSelect = GTM_CONFIGURABLE_CLOCK_4 in GtmTimerConfiguration_0
	
	In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITHOUT_DIV_SEL1
		`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel CM0 value*/`
		`500000U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Timer Channel CM0 value*/`
		`400000U,`
		`}`

Configure GtmClusterConfClock4Src= CMU_CONF_CLOCK8_SEL1 in GtmGlobalConfiguration/*[1]/GtmClusterConf/ GtmClusterConf_0/ GtmClusterConfClockSetting In WdgSettingsConfig_0/WdgTriggerTimerSetting Configure WdgSlowRefreshTime=0.005 WdgFastRefreshTime=0.004	

Configure GtmTimerUsed = McuGtmTomAllocationConf_1/McuGtmTomChannelAllocationConf_1 GtmTimerClockSelect = GTM_FIXED_CLOCK_2 in GtmTimerConfiguration_0
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITHOUT_DIV_SEL1
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_1/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITH_DIV_SEL2
 	
Configure GtmCmuFixedClockSel = CMU_GLOBAL_CLOCK_SEL0 in GtmGlobalConfiguration_0/ GtmFixedClockSetting
 	
In WdgSettingsConfig_0/WdgTriggerTimerSetting
Configure WdgSlowRefreshTime=0.007
WdgFastRefreshTime=0.004
	`/*GTM channel structure for Slow*/`
	`{`
	`  /*Timer Channel CM0 value*/`
	`  1367U,`
	`}`
	`/*GTM channel structure for Fast*/`
	`{`
	`  /*Timer Channel CM0 value*/`
	`  781U,`
	`}`

Configure GtmTimerUsed = McuGtmTomAllocationConf_1/McuGtmTomChannelAllocationConf_1
GtmTimerClockSelect = GTM_FIXED_CLOCK_0 in GtmTimerConfiguration_0
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITHOUT_DIV_SEL1
 	
Configure GtmCmuGlobalClockNumerator=100 GtmCmuGlobalClockDenominator =1 in GtmGlobalConfiguration/*[1]/McuGtmClockManagementConf/
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_1/ GtmCmuClusterInputClockDividerEnable= CLS_CLK_CFG_ENABLED_WITH_DIV_SEL2
 	
Configure GtmCmuFixedClockSel = CMU_GLOBAL_CLOCK_SEL0 in GtmGlobalConfiguration_0/ GtmFixedClockSetting
 	
In WdgSettingsConfig_0/WdgTriggerTimerSetting
Configure WdgSlowRefreshTime=0.007 WdgFastRefreshTime=0.004
	`/*GTM channel structure for Slow*/`
	`{`
	`/*Timer Channel CM0 value*/`
	`3500U,`
	`}`
	`/*GTM channel structure for Fast*/`
	`{`
	`/*Timer Channel CM0 value*/`
	`2000U,`
	`}`

1.2.2.6 Member: TimerChCM1Reg

Table 40 TimerChCM1Reg

Name	TimerChCM1Reg
Type	uint32

Description	TOM/ATOM channel CM1 register value.	
Verification method	The structure member is generated as value of the CM1 register for TOM/ATOM channel. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel CM1 value*/`
		`0x0U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Timer Channel CM1 value*/`
		`0x0U,`
		`}`

1.2.2.7 Member: TimerChSR0Reg

Table 41 TimerChSR0Reg

Name	TimerChSR0Reg	
Type	uint32	
Description	TOM/ATOM channel SR0 register value.	
Verification method	The structure member is generated as value of the SR0 register for TOM/ATOM channel. Steps to calculate TimerChCM0Reg • GTM frequency calculation fGtm=((McuGTMFrequency * GtmDenominator)/ GtmNumerator) fGtm=(fGtm / GtmClusterDivVal) fGtm= fGtm/ ClockDivider • TimerChCM0Reg value is calculated based on RefreshTime. Refresh time can be SlowRefreshTime/ FastRefreshTime depending on the mode of the watchdog. TimerChCM0Reg = ((RefreshTime*1000) * fGtm)/1000)	
Example(s)	**Action**	**Generated output**

	Configure GtmTimerUsed = McuGtmAtomAllocationConf_0 / McuGtmAtomChannelAllocationConf_0 and GtmTimerClockSelect =	`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel CM0 value*/`

GTM_CONFIGURABLE_CLOCK_4
in GtmTimerConfiguration_0
GTM frequency = 50MHZ
 	
In GtmGlobalConfiguration_0/
GtmClusterConf/ GtmClusterConf_0/
GtmCmuClusterInputClockDivid
erEnable=
CLS_CLK_CFG_ENABLED_WITH_
DIV_SEL2
 	
Configure GtmClusterConfClock4Src=
CMU_CONF_CLOCK4_SEL0 in
GtmGlobalConfiguration/*[1]/GtmCluster
Conf/ GtmClusterConf_1/
GtmClusterConfClockSetting
 	
In WdgSettingsConfig_0/WdgTrigg
erTimerSetting
Configure WdgSlowRefreshTime=0.33
WdgFastRefreshTime=0.11
	16500000U,
	}
	/*GTM channel structure for Fast*/
	{
	/*Timer Channel CM0 value*/
	5500000U,
	}

Configure GtmTimerUsed =
McuGtmAtomAllocationConf_0
/ McuGtmAtomChannelAllocatio
nConf_0 and
GtmTimerClockSelect =
GTM_CONFIGURABLE_CLOCK_4
in GtmTimerConfiguration_0
 	
In GtmGlobalConfiguration_0/
GtmClusterConf/ GtmClusterConf_0/
GtmCmuClusterInputClockDivid
erEnable=
CLS_CLK_CFG_ENABLED_WITHO
UT_DIV_SEL1
 	
Configure GtmClusterConfClock4Src=
CMU_CONF_CLOCK8_SEL1 in
GtmGlobalConfiguration/*[1]/GtmClusterCo
nf/ GtmClusterConf_0/
	/*GTM channel structure for Slow*/
	{
	/*Timer Channel CM0 value*/
	500000U,
	}
	/*GTM channel structure for Fast*/
	{
	/*Timer Channel CM0 value*/
	400000U,
	}

GtmClusterConfClockSetting In WdgSettingsConfig_0/WdgTrigg erTimerSetting Configure WdgSlowRefreshTime=0.005 WdgFastRefreshTime=0.004	

Configure GtmTimerUsed = McuGtmTomAllocationConf_1/McuGtm TomChannelAllocationConf_1 GtmTimerClockSelect = GTM_FIXED_CLOCK_2 in GtmTimerConfiguration_0
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDivid erEnable= CLS_CLK_CFG_ENABLED_WITHO UT_DIV_SEL1
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_1/ GtmCmuClusterInputClockDivid erEnable= CLS_CLK_CFG_ENABLED_WITH_DIV _SEL2
 	
Configure GtmCmuFixedClockSel = CMU_GLOBAL_CLOCK_SEL0 in GtmGlobalConfiguration_0/ GtmFixedClockSetting
 	
In WdgSettingsConfig_0/WdgTrigg erTimerSetting
Configure WdgSlowRefreshTime=0.007 WdgFastRefreshTime=0.004
	`/*GTM channel structure for Slow*/`
	`{`
	`/*Timer Channel CM0 value*/`
	`1367U,`
	`}`
	`/*GTM channel structure for Fast*/`
	`{`
	`/*Timer Channel CM0 value*/`
	`781U,`
	`}`

Configure GtmTimerUsed = McuGtmTomAllocationConf_1/McuGtm TomChannelAllocationConf_1 GtmTimerClockSelect =	`/*GTM channel structure for Slow*/`
	`{`
	`/*Timer Channel CM0 value*/`

GTM_FIXED_CLOCK_0 in GtmTimerConfiguration_0
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_0/ GtmCmuClusterInputClockDivid erEnable= CLS_CLK_CFG_ENABLED_WITHO UT_DIV_SEL1
 	
Configure GtmCmuGlobalClockNumerator=100 GtmCmuGlobalClockDenominator =1 in GtmGlobalConfiguration/*[1]/McuGtmClockMa nagementConf/
 	
In GtmGlobalConfiguration_0/ GtmClusterConf/ GtmClusterConf_1/ GtmCmuClusterInputClockDivid erEnable= CLS_CLK_CFG_ENABLED_WITH_DIV _SEL2
 	
Configure GtmCmuFixedClockSel = CMU_GLOBAL_CLOCK_SEL0 in GtmGlobalConfiguration_0/ GtmFixedClockSetting
 	
In WdgSettingsConfig_0/WdgTrigg erTimerSetting
Configure WdgSlowRefreshTime=0.007 WdgFastRefreshTime=0.004
	3500U,
	}
	/*GTM channel structure for Fast*/
	{
	/*Timer Channel CM0 value*/
	2000U,
	}

1.2.2.8 Member: TimerChSR1Reg

Table 42 TimerChSR1Reg

Name	TimerChSR1Reg
Type	uint32
Description	TOM/ATOM channel SR1 register value.
Verification method	The structure member is generated as value of the SR1 register for TOM/ATOM

	channel. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel SR1 value*/`
		`0x0U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Timer Channel SR1 value*/`
		`0x0U,`
		`}`

1.2.2.9 Member: TimerChPortOutConfig

Table 43 TimerChPortOutConfig

Name	TimerChPortOutConfig	
Type	uint32	
Description	TOM/ATOM to port configuration.	
Verification method	The structure member is generated as value of the port configuration for TOM/ATOM . *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`/*GTM channel structure for Slow*/`
		`{`
		`/*Channel to Port Value*/`
		`0x0U,`
		`}`
		`/*GTM channel structure for Fast*/`
		`{`
		`/*Channel to Port Value*/`
		`0x0U,`
		`}`

1.2.2.10 Member: TimerChIntEnMode

Table 44 TimerChIntEnMode

Name	TimerChIntEnMode	
Type	uint8	
Description	TOM/ATOM channel interrupt enable and interrupt mode values.	
Verification method	The structure member is generated as value of the interrupt enable and interrupt mode for TOM/ATOM . *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`/*GTM channel structure for Slow*/`
		`{`
		`/*Timer Channel Interrupt Enable`

		value*/
		0x81U,
		}
		/*GTM channel structure for Fast*/
		{
		/*Timer Channel Interrupt Enable value*/
		0x81U,
		}

1.2.3 Structure: Wdg_StmConfig_ <x>[_<variant>]

Table 45 Wdg_StmConfig_ <x>[_<variant>]

Name	Wdg_StmConfig_ <x>[_<variant>]	
Type	Mcu_17_Stm_TimerConfigType	
Description	Contains STM timer configuration information.	
Verification method	The generated file has this structure if STM timer is selected as WdgTriggerTimerSelection assigned to Core <x>. <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 WdgTriggerTimerSelection = STM_TIMER
	Allocate STM TIMER 1 in Resource Manager to Core0
	In MCU allocate STM CMP0 to Watchdog.
	(variant-unaware)
		static const Mcu_17_Stm_TimerConfigType Wdg_StmConfig_0 =
		{
		/*STM compare Reg */
		0x00000000U,
		/* StmTimerId*/
		0x1U,
		/*Cmp Register Id*/
		0x0U,
		/* Value for the CMCON register */
		0x1FU,
		/* Reserved */
		0x00U
		};

	Configure WdgTriggerTimerSelection = STM_TIMER Allocate STM TIMER 4 in Resource Manager to Core0 In MCU allocate STM CMP1 to Watchdog.	static const Mcu_17_Stm_TimerConfigType Wdg_StmConfig_0_Petrol = { /*STM compare Reg */ 0x00000000U, /* StmTimerId*/

	(variant-aware. Variant name is 'Petrol')	0x4U, /*Cmp Register Id*/ 0x1U, /* Value for the CMCON register */ 0x1FU, /* Reserved */ 0x00U };

1.2.3.1 Member: CompareRegVal

Table 46 CompareRegVal

Name	CompareRegVal	
Type	uint32	
Description	Compare register value for STM timer.	
Verification method	The structure member is generated as value of the compare register for STM timer. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	{
		/*STM compare Reg */
		0x00000000U,
		}

1.2.3.2 Member: StmTimerId

Table 47 StmTimerId

Name	StmTimerId	
Type	unsigned_int	
Description	STM timer number configured for Core.	
Verification method	The structure member is generated as value of STM timer configured to Core.	
Example(s)	**Action**	**Generated output**

	Configure WdgTriggerTimerSelection = STM_TIMER
	Allocate STM TIMER 4 in Resource Manager to Core0
		{
		/* StmTimerId*/
		0x4U,
		}

1.2.3.3 Member: CMPRegId

Table 48 CMPRegId

Name	CMPRegId	
Type	unsigned_int	
Description	Compare register for STM timer.	
Verification method	The structure member is generated as value of compare register used for STM .	
Example(s)	**Action**	**Generated output**

	Configure WdgTriggerTimerSelection = STM_TIMER
	Allocate STM TIMER 4 in Resource Manager to Core0
	
	In MCU allocate STM CMP1 to Watchdog.
		{
		/*Cmp Register Id*/
		0x1U,
		}

	Configure WdgTriggerTimerSelection = STM_TIMER
	Allocate STM TIMER 4 in Resource Manager to Core0
	
	In MCU allocate STM CMP0 to Watchdog.
		{
		/*Cmp Register Id*/
		0x0U,
		}

1.2.3.4 Member: CmconRegVal

Table 49 CmconRegVal

Name	CmconRegVal	
Type	unsigned_int	
Description	Compare match control register value.	
Verification method	The structure member is generated as value of the compare match control register for STM timer. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	{
		/* Value for the CMCON register */
		0x1FU,
		}

1.2.3.5 Member: reserved

Table 50 reserved

Name	reserved	
Type	unsigned_int	
Description	Reserved.	
Verification method	The structure member is generated as value zero used as reserved value. *Note: This macro is not configurable by the user*	
Example(s)	**Action**	**Generated output**

	Generate configuration file Wdg_17_Scu[_<variant>]_PBcfg.c	`{`
		`  /* Reserved */`
		`  0x00U`
		`}`

1.3 File: Wdg_17_Scu[_<variant>]_PBcfg.h

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

1.3.1 Structure: Wdg_17_Scu_Config_<x>[_<variant>]

Table 1 Wdg_17_Scu_Config_<x>[_<varaint>]

Name	Wdg_17_Scu_Config_<x>[_<variant>]	
Type	Wdg_17_Scu_ConfigType	
Description	Root configuration structure of WDG driver which will be used during core specific initialization. (x in the range of 0 to 5)	
Verification method	The generated structure is present in Wdg_17_Scu[_<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 WDG for Core 0 (variant-unaware)	`extern const Wdg_17_Scu_ConfigType Wdg_17_Scu_Config_0;`
	Configure WDG for Core 2 (variant-aware. Variant name is 'Petrol')	`extern const Wdg_17_Scu_ConfigType Wdg_17_Scu_Config_2_Petrol;`

Revision history

Major changes since the last revision

Date	Version	Description
2023-05-19	V6.0	Document Released.
2023-05-18	V5.1	Documentation updated to change DEM to Productions error where applicable in sections 1.1.17, 1.1.18, 1.1.19, 1.1.20
2020-10-19	v5.0	Document Released
2020-10-16	v4.1	Wdg_17_Scu driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_BASIC.pdf to this document
2019-07-19	v4.0	Updated the version and revision history. No other changes identified compared to the previous version.
2019-02-27	v1.10.0_3.0	Added Pbcfg.h
2019-02-25	v1.10.0_2.0	Section 1.2.2.5, 1.2.2.7 updated for JIRA 0000053912-5573
2019-02-20	v1.10.0_1.0	Initial Version

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

Edition 2023-05-19
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.