User Manual

for MPC574XG MCU Driver

Document Number: UM35MCUASR4.2 Rev0002R1.0.0
Rev. 1.0

Contents

Section number **Title** **Page**

Chapter 1 Revision History

Chapter 2 Introduction

Chapter 3 Driver

Section number Title Page

Section number **Title** **Page**

Chapter 4
Tresos Configuration Plug-in

Section number Title Page

Section number **Title** **Page**

Section number Title Page

Section number	Title	Page

Section number	Title	Page

Section number Title Page

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	10/02/2017	Dang Cong	Update for MPC574XG/C 4.2 RTM 1.0.0 Release

Chapter 2
Introduction

This User Manual describes NXP Semiconductor AUTOSAR Micro Controller Unit (MCU) for MPC574XG .

AUTOSAR MCU driver configuration parameters and deviations from the specification are described in MCU Driver chapter of this document. AUTOSAR MCU driver requirements and APIs are described in the AUTOSAR MCU driver software specification document.

2.1 Supported Derivatives

The software described in this document is intented to be used with the following microcontroller devices of NXP Semiconductor .

Table 2-1. MPC574XG Derivatives

NXP Semiconductor	MPC5748G_LQFP176, MPC5748G_MAPBGA256, MPC5748G_MAPBGA324, MPC5747G_LQFP176, MPC5747G_MAPBGA256, MPC5747G_MAPBGA324, MPC5746G_LQFP176, MPC5746G_MAPBGA256, MPC5746G_MAPBGA324, MPC5748C_LQFP176, MPC5748C_MAPBGA256, MPC5748C_MAPBGA324, MPC5747C_LQFP176, MPC5747C_MAPBGA256, MPC5747C_MAPBGA324, MPC5746C_LQFP176, MPC5746C_MAPBGA256, MPC5746C_MAPBGA324, MPC5746C_MAPBGA100, MPC5745C_LQFP176, MPC5745C_MAPBGA256, MPC5745C_MAPBGA100, MPC5744C_LQFP176, MPC5744C_MAPBGA256,

Table 2-1. MPC574XG Derivatives

	MPC5744C_MAPBGA100, MPC5746B_LQFP176, MPC5746B_MAPBGA256, MPC5746B_MAPBGA100, MPC5744B_LQFP176, MPC5744B_MAPBGA256, MPC5744B_MAPBGA100, MPC5745B_LQFP176, MPC5745B_MAPBGA256, MPC5745B_MAPBGA100

All of the above microcontroller devices are collectively named as MPC574XG .

2.2 Overview

AUTOSAR (AUTomotive Open System ARchitecture) is an industry partnership working to establish standards for software interfaces and software modules for automobile electronic control systems.

AUTOSAR

- paves the way for innovative electronic systems that further improve performance, safety and environmental friendliness.
- is a strong global partnership that creates one common standard: "Cooperate on standards, compete on implementation".
- is a key enabling technology to manage the growing electrics/electronics complexity. It aims to be prepared for the upcoming technologies and to improve cost-efficiency without making any compromise with respect to quality.
- facilitates the exchange and update of software and hardware over the service life of the vehicle.

2.3 About this Manual

This Technical Reference employs the following typographical conventions:

Boldface type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

This is a note.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
DEM	Diagnostic Event Manager
DET	Default Error Tracer
C/CPP	C and C++ Source Code
VLE	Variable Length Encoding
PCS	Progressive Clock Switching
CMU	Clock Monitoring Unit
N/A	Not Applicable
MCU	Micro Controller Unit

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 4.2 Rev0002MCU Driver Software Specification Document.	R4.2.2
2	MPC5748G Reference Manual	Rev. 5, 12/2016
3	MPC5746C Reference Manual	Rev. 4, 12/2016
4	MPC5748G_1N81M_Rev.2 (official document) (1N81M)	Jun-16
5	MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0 (internal document) (1N81M, 0N78S)	31.10.2016
6	MPC5746C_1N06M_Rev.4 (official document) (1N06M)	Jul-16
7	MPC5746C_cut1.1_cut2.0_cut2.1_comparison_v0 (internal document) (1N06M, 0N84S, 1N84S)	14-Sep-16
8	C3M_cut2.1_new_errata_20170113 (internal document) (1N84S)	13-Jan-17

Chapter 3
Driver

3.1 Requirements

Requirements for this driver are detailed in the AUTOSAR 4.2 Rev0002MCU Driver Software Specification document (See Table Reference List).

3.2 Driver Design Summary

The MPC5748G contains the following blocks:

- IPV_CMU
- IPV_MCV4
- IPV_EMIOS
- IPV_FLASHV2
- IPV_PRAM
- IPV_STCUV2
- IPV_SSCM
- IPV_PPLDIG
- IPV_SIUL2
- IPV_FIRC
- IPV_FXOSC
- IPV_SIRC
- IPV_SXOSC
- IPV_PMCDIG
- IPV_LPU

IPV_CMU measures and monitors the frequency of clock sources.

The **IPV_CMU** covers the following IP's:

- **CMU**Clock Monitor Unit

IPV_MCV4 controlls the clock sources, clock tree, reset generation, power control and clock monitoring.

The **IPV_MCV4** covers the following IP's:

- **MC_CGM**Clock generation and distribution module.
- **MC_RGM**Reset generation and control module.
- **PCU** Power control unit
- **MC_ME** Power mode control

IPV_EMIOS is a timer module.

The **IPV_EMIOS** covers the following IP's:

- **eMIOS**Configurable Enhanced Modular IO Subsystem

IPV_FLASHV2 controlls the on-chip flash wait states, prefetch and access control.

The **IPV_FLASHV2** covers the following IP's:

- **PFLASH** Flash wait states, prefetch and access control

IPV_PRAM controlls the on-chip sram operation.

The **IPV_PRAM** covers the following IP's:

- **PRAM** Internal SRAM operation

IPV_STCUV2 receives information regarding the SRAM status.

The **IPV_STCUV2** covers the following IP's:

- **STCU** Internal SRAM status

IPV_SSCM receives information regarding the system status.

The **IPV_SSCM** covers the following IP's:

- **SSCM**system configuration

IPV_SIUL2 receives information regarding the system configuration.

The **IPV_SIUL2** covers the following IP's:

- **SIUL2** system configuration

IPV_PLLDIG controls PLL clock generation modules.

The **IPV_PLLDIG** covers the following IP's:

- **PLLDIG0** PLL0

IPV_FXOSC controls the external oscillator (8 - 40 MHz).

The **IPV_FXOSC** covers the following IP's:

- **FXOSC** FXOSC

IPV_SXOSC provides a 32 KHz clock with control and status registers accessible for application use.

The **IPV_SXOSC** covers the following IP's:

- **SXOSC** SXOSC

IPV_FIRC provides the control of internal RC oscillator (16 MHz).

The **IPV_FIRC** covers the following IP's:

- **FIRC** FIRC

IPV_SIRC provides the control of internal RC oscillator (128 kHz).

The **IPV_SIRC** covers the following IP's:

- **SIRC** SIRC

IPV_PMCDIG provides the Power Management Controller digital interface.

The **IPV_PMCDIG** covers the following IP's:

- **PMCDIG** PMCDIG

IPV_LPU provides LPU configuration and control.

The **IPV_LPU** covers the following IP's:

- **LPU** LPU

3.3 Hardware Resources

The hardware configured by the Mcu driver is the same between derivatives.

3.4 Deviation from Requirements

As a deviation from standard:

- Mcuc_PBcfg_**VariantNo**.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, PB)

- Mcu_Cfg.c file will contain the definition for all parameters that are not variant aware

The driver deviates from the AUTOSAR MCU Driver software specification in some places. The table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the MCU Driver. Table Table 3-1 provides Status column description.

Table 3-1. Deviations Status Column Description

Term	Definition
N/A	Not available
N/T	Not testable
N/S	Out of scope
N/I	Not implemented
N/F	Not fully implemented

Below table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, or out of scope for the driver.

Table 3-2. Driver Deviations Table

Requirement	Status	Description	Notes
SWS_Mcu_00053	N/A	The requirement is refering to a clock failure error that is detected via a clock monitoring unit (CMU) interrupt. Because of PR-MCAL-3324.mcu which states that DEM calls must not happen in interrupt context a new requirement was created CPR-MCAL-733.mcu (The MCU module shall provide a configurable user callback for the case when an error is detected via the error ISRs) to define a new type of error that can be called via a user configurable callback. For the clock failure case the error MCU_E_ISR_CLOCK_FAILURE was defined and if the error notification is configured by the application the CMU interrupt will report it.	
SWS_Mcu_00257	N/A	The requirement is refering to a clock failure error that is detected via a clock monitoring unit (CMU) interrupt. Because of PR-	

Table continues on the next page...

Table 3-2. Driver Deviations Table (continued)

Requirement	Status	Description	Notes
		MCAL-3324.mcu which states that DEM calls must not happen in interrupt context a new requirement was created CPR-MCAL-733.mcu (The MCU module shall provide a configurable user callback for the case when an error is detected via the error ISRs) to define a new type of error that can be called via a user configurable callback. For the clock failure case the error MCU_E_ISR_CLOCK_FAILURE was defined and if the error notification is configured by the application the CMU interrupt will report it.	
SWS_Mcu_00258	N/A	The requirement is refering to a clock failure error that is detected via a clock monitoring unit (CMU) interrupt. Because of PR-MCAL-3324.mcu which states that DEM calls must not happen in interrupt context a new requirement was created CPR-MCAL-733.mcu (The MCU module shall provide a configurable user callback for the case when an error is detected via the error ISRs) to define a new type of error that can be called via a user configurable callback. For the clock failure case the error MCU_E_ISR_CLOCK_FAILURE was defined and if the error notification is configured by the application the CMU interrupt will report it.	

3.5 Driver Limitation

3.5.1

None

3.6 Driver usage and configuration tips

3.6.1

For reconfiguring the PLLs using Mcu_InitClock and Mcu_DistributePllClock the peripherals that are clocked using the PLL that needs to be reconfigured should be turned OFF using Mcu_SetMode to transition in a mode where that peripheral is OFF.

For bypassing the configuration of a clock source, the system clock or of auxiliary clocks during Mcu_InitClock the check box "[source] under MCU control" should be unchecked. This will generate smaller configurations that will be updated faster and more efficiently.

For using STANDBY mode configuration on Calypso 3M, please pay attention to errata ERR009139 (SRAM contents are not guaranteed in STANDBY Modes).

To enable SRAM retention in STANDBY do the step following (step1 and step2 were supported by MCU driver):

Step 1: Enable GPR_SLEEP_BIT prior to STANDBY entry (no SRAM access is permitted after this write)

Step 2: Perform the mode change to enter into STANBDY Mode

Step 3: On wakeup from STANDBY, clear the GPR_SLEEP_BIT before any access is made to the SRAM

Step 3 must be done in the startup code of the application and bits 25, 30, 31 must be configured in the register GPR_SLEEP (This register defines the sleep status of SRAM memory blocks) at the address 0xFFF94014

Example of code that will clear the bits is: (*(volatile uint32*)0xFFF94014UL) &= ~(0x00000043UL)

3.7 Runtime Errors

The driver generates the following errors at runtime.

Table 3-3. Default Errors (reported by DET)

Function	Error Code	Condition triggering the error
Mcu_Init	MCU_E_INIT_FAILED	Invalid configuration pointer
Mcu_InitClock	MCU_E_PARAM_CLOCK	Invalid input parameter
Mcu_SetMode	MCU_E_PARAM_MODE	Invalid input parameter

Table continues on the next page...

Table 3-3. Default Errors (reported by DET) (continued)

Function	Error Code	Condition triggering the error
Mcu_InitRamSection	MCU_E_PARAM_RAMSECTION	Invalid input parameter or invalid memory configuration
Mcu_DistributePllClock	MCU_E_PLL_NOT_LOCKED	One of the used PLL's failed to achieve lock
All functions, except Mcu_Init and Mcu_GetVersionInfo	MCU_E_UNINIT	The driver is in an uninitialized state
Mcu_GetMidrStructure	MCU_E_PARAM_POINTER	Invalid input parameter
Mcu_GetVersionInfo	MCU_E_PARAM_POINTER	Invalid input parameter
Mcu_Init	MCU_E_ALLREADY_INITIALIZED	The driver is already initialized
Mcu_DisableCmu	MCU_E_CMU_INDEX_OUT_OF_RANGE	Invalid input parameter

Table 3-4. Production Errors (reported by DEM)

Function	Error Code	Condition triggering the error
Mcu_GetResetReason	Mcu_E_TimeoutFailure	Reset flags could not be cleared.
Mcu_GetResetRawValue	Mcu_E_TimeoutFailure	Reset flags could not be cleared.
Mcu_SetMode	Mcu_E_TimeoutFailure	The MC_ME or LPU mode transition failed.
Mcu_Init	Mcu_E_TimeoutFailure	The MC_ME mode transition failed.
Mcu_InitClock	Mcu_E_TimeoutFailure	The MC_ME mode transition failed.

3.8 Software specification

The following sections contains driver software specifications.

3.8.1 Define Reference

Constants supported by the driver are as per AUTOSAR MCU Driver software specification Version 4.2 Rev0002 .

3.8.1.1 Define MCU_INSTANCE_ID

Table 3-5. Define MCU_INSTANCE_ID Description

Name	MCU_INSTANCE_ID
Initializer	(uint8)0x0U

3.8.1.2 Define MCU_MODULE_ID

Table 3-6. Define MCU_MODULE_ID Description

Name	MCU_MODULE_ID
Initializer	101

3.8.1.3 Define MCU_INIT_ID

Service Ids for MCU APIs.

Table 3-7. Define MCU_INIT_ID Description

Name	MCU_INIT_ID
Initializer	(uint8)0x00U

3.8.1.4 Define MCU_INITRAMSECTION_ID

Service Ids for MCU APIs.

Table 3-8. Define MCU_INITRAMSECTION_ID Description

Name	MCU_INITRAMSECTION_ID
Initializer	(uint8)0x01U

3.8.1.5 Define MCU_INITCLOCK_ID

Service Ids for MCU APIs.

Table 3-9. Define MCU_INITCLOCK_ID Description

Name	MCU_INITCLOCK_ID
Initializer	(uint8)0x02U

3.8.1.6 Define MCU_DISTRIBUTEPLLCLOCK_ID

Service Ids for MCU APIs.

Table 3-10. Define MCU_DISTRIBUTEPLLCLOCK_ID Description

Name	MCU_DISTRIBUTEPLLCLOCK_ID
Initializer	(uint8)0x03U

3.8.1.7 Define MCU_GETPLLSTATUS_ID

Service Ids for MCU APIs.

Table 3-11. Define MCU_GETPLLSTATUS_ID Description

Name	MCU_GETPLLSTATUS_ID
Initializer	(uint8)0x04U

3.8.1.8 Define MCU_GETRESETREASON_ID

Service Ids for MCU APIs.

Table 3-12. Define MCU_GETRESETREASON_ID Description

Name	MCU_GETRESETREASON_ID
Initializer	(uint8)0x05U

3.8.1.9 Define MCU_GETRESETRAWVALUE_ID

Service Ids for MCU APIs.

Table 3-13. Define MCU_GETRESETRAWVALUE_ID Description

Name	MCU_GETRESETRAWVALUE_ID
Initializer	(uint8)0x06U

3.8.1.10 Define MCU_PERFORMRESET_ID

Service Ids for MCU APIs.

Table 3-14. Define MCU_PERFORMRESET_ID Description

Name	MCU_PERFORMRESET_ID
Initializer	(uint8)0x07U

3.8.1.11 Define MCU_SETMODE_ID

Service Ids for MCU APIs.

Table 3-15. Define MCU_SETMODE_ID Description

Name	MCU_SETMODE_ID
Initializer	(uint8)0x08U

3.8.1.12 Define MCU_GETVERSIONINFO_ID

Service Ids for MCU APIs.

Table 3-16. Define MCU_GETVERSIONINFO_ID Description

Name	MCU_GETVERSIONINFO_ID
Initializer	(uint8)0x09U

3.8.1.13 Define MCU_GETRAMSTATE_ID

Service Ids for MCU APIs.

Table 3-17. Define MCU_GETRAMSTATE_ID Description

Name	MCU_GETRAMSTATE_ID
Initializer	(uint8)0x0AU

3.8.1.14 Define MCU_GETPOWERDOMAIN_ID

Service Ids for MCU APIs.

Table 3-18. Define MCU_GETPOWERDOMAIN_ID Description

Name	MCU_GETPOWERDOMAIN_ID
Initializer	(uint8)0x0BU

3.8.1.15 Define MCU_GETPERIPHERALSTATE_ID

Service Ids for MCU APIs.

Table 3-19. Define MCU_GETPERIPHERALSTATE_ID Description

Name	MCU_GETPERIPHERALSTATE_ID
Initializer	(uint8)0x0CU

3.8.1.16 Define MCU_GETSYSTEMSTATE_ID

Service Ids for MCU APIs.

Table 3-20. Define MCU_GETSYSTEMSTATE_ID Description

Name	MCU_GETSYSTEMSTATE_ID
Initializer	(uint8)0x0DU

3.8.1.17 Define MCU_GETPOWERMODESTATE_ID

Service Ids for MCU APIs.

Table 3-21. Define MCU_GETPOWERMODESTATE_ID Description

Name	MCU_GETPOWERMODESTATE_ID
Initializer	(uint8)0x0EU

3.8.1.18 Define MCU_GETMEMCONFIG_ID

Service Ids for MCU APIs.

Table 3-22. Define MCU_GETMEMCONFIG_ID Description

Name	MCU_GETMEMCONFIG_ID
Initializer	(uint8)0x13U

3.8.1.19 Define MCU_SSCMGETSTATUS_ID

Table 3-23. Define MCU_SSCMGETSTATUS_ID Description

Name	MCU_SSCMGETSTATUS_ID
Initializer	((uint8)0x15U)

3.8.1.20 Define MCU_SSCMGETUOPS_ID

Service Ids for MCU APIs.

Table 3-24. Define MCU_SSCMGETUOPS_ID Description

Name	MCU_SSCMGETUOPS_ID
Initializer	((uint8)0x16U)

3.8.1.21 Define MCU_DISABLECMU_ID

Service Ids for MCU APIs.

Table 3-25. Define MCU_DISABLECMU_ID Description

Name	MCU_DISABLECMU_ID
Initializer	((uint8)0x17U)

3.8.1.22 Define MCU_GETMIDRSTRUCTURE_ID

Service Ids for MCU APIs.

Table 3-26. Define MCU_GETMIDRSTRUCTURE_ID Description

Name	MCU_GETMIDRSTRUCTURE_ID

Table continues on the next page...

Table 3-26. Define MCU_GETMIDRSTRUCTURE_ID Description (continued)

Initializer	(uint8)0x14U

3.8.1.23 Define MCU_EMIOSCONFIGUREGPREN_ID

The function Mcu_EmiosConfigureGpren is only available if the runtime parameter McuEmiosConfigureGprenApi is set to TRUE.

Satisfied Requirements: PR-MCAL-3298.mcu

Table 3-27. Define MCU_EMIOSCONFIGUREGPREN_ID Description

Name	MCU_EMIOSCONFIGUREGPREN_ID
Initializer	(0x16U)

3.8.1.24 Define MCU_E_PARAM_CONFIG

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

If development error detection is enabled, the parameter ConfigPtr shall be checked for being NULL. If the parmameter is NULL, the error code MCU_E_PARAM_CONFIG shall be reported to the DET.

Table 3-28. Define MCU_E_PARAM_CONFIG Description

Name	MCU_E_PARAM_CONFIG
Initializer	((uint8)0x0AU)

3.8.1.25 Define MCU_E_PARAM_CLOCK

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

The ClockSetting shall be within the settings defined in the configuration data structure. If not, the error code MCU_E_PARAM_CLOCK shall be reported to the DET.

Table 3-29. Define MCU_E_PARAM_CLOCK Description

Name	MCU_E_PARAM_CLOCK
Initializer	((uint8)0x0BU)

3.8.1.26 Define MCU_E_PARAM_MODE

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

McuMode shall be within the modes defined in the configuration data structure. If not, the error code MCU_E_PARAM_MODE shall be reported to the DET.

Table 3-30. Define MCU_E_PARAM_MODE Description

Name	MCU_E_PARAM_MODE
Initializer	((uint8)0x0CU)

3.8.1.27 Define MCU_E_PARAM_RAMSECTION

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

RamSection shall be within the sections defined in the configuration data structure. If not, the error code MCU_E_PARAM_RAMSECTION shall be reported to the DET.

Table 3-31. Define MCU_E_PARAM_RAMSECTION Description

Name	MCU_E_PARAM_RAMSECTION
Initializer	((uint8)0x0DU)

3.8.1.28 Define MCU_E_PLL_NOT_LOCKED

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

The error shall be reported if the status of the PLL is detected as not locked with the function Mcu_DistributePllClock() .

Table 3-32. Define MCU_E_PLL_NOT_LOCKED Description

Name	MCU_E_PLL_NOT_LOCKED
Initializer	((uint8)0x0EU)

3.8.1.29 Define MCU_E_UNINIT

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

If development error detection is enabled and if any other function (except Mcu_GetVersionInfo) of the MCU module is called before Mcu_Init function, the error code MCU_E_UNINIT shall be reported to the DET.

Table 3-33. Define MCU_E_UNINIT Description

Name	MCU_E_UNINIT
Initializer	((uint8)0x0FU)

3.8.1.30 Define MCU_E_PARAM_POINTER

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

if development error detection is enabled, the parameter versioninfo shall be checked for being NULL. The error MCU_E_PARAM_POINTER shall be reported in case the value is a NULL pointer.

Table 3-34. Define MCU_E_PARAM_POINTER Description

Name	MCU_E_PARAM_POINTER
Initializer	((uint8)0x10U)

3.8.1.31 Define MCU_E_CLOCK_UNINIT

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-35. Define MCU_E_CLOCK_UNINIT Description

Name	MCU_E_CLOCK_UNINIT
Initializer	(uint8)0x11U

3.8.1.32 Define MCU_E_SWITCH_MODE_INVALID

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-36. Define MCU_E_SWITCH_MODE_INVALID Description

Name	MCU_E_SWITCH_MODE_INVALID
Initializer	((uint8)0x12U)

3.8.1.33 Define MCU_E_ALLREADY_INITIALIZED

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-37. Define MCU_E_ALLREADY_INITIALIZED Description

Name	MCU_E_ALLREADY_INITIALIZED
Initializer	(uint8)0x13U

3.8.1.34 Define MCU_E_TRANSITION

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-38. Define MCU_E_TRANSITION Description

Name	MCU_E_TRANSITION
Initializer	((uint8)0x20U)

3.8.1.35 Define MCU_E_EMIOS_DEACTIVATED

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-39. Define MCU_E_EMIOS_DEACTIVATED Description

Name	MCU_E_EMIOS_DEACTIVATED
Initializer	(uint8)0x21U

3.8.1.36 Define MCU_E_CMU_INDEX_OUT_OF_RANGE

Development error values are of type uint8. The following errors and exceptions shall be detectable by the MCU module depending on its build version (development/production mode).

Table 3-40. Define MCU_E_CMU_INDEX_OUT_OF_RANGE Description

Name	MCU_E_CMU_INDEX_OUT_OF_RANGE
Initializer	((uint8)0x22U)

3.8.1.37 Define MCU_E_ISR_CLOCK_FAILURE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-41. Define MCU_E_ISR_CLOCK_FAILURE Description

Name	MCU_E_ISR_CLOCK_FAILURE
Initializer	(uint8)0x01U

3.8.1.38 Define MCU_E_ISR_PLL_LOCK_FAILURE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-42. Define MCU_E_ISR_PLL_LOCK_FAILURE Description

Name	MCU_E_ISR_PLL_LOCK_FAILURE
Initializer	((uint8)0x02U)

3.8.1.39 Define MCU_E_ISR_SAFE_MODE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-43. Define MCU_E_ISR_SAFE_MODE Description

Name	MCU_E_ISR_SAFE_MODE
Initializer	((uint8)0x03U)

3.8.1.40 Define MCU_E_ISR_INVALID_MODE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-44. Define MCU_E_ISR_INVALID_MODE Description

Name	MCU_E_ISR_INVALID_MODE
Initializer	((uint8)0x04U)

3.8.1.41 Define MCU_E_ISR_INVALID_MODE_CONFIG

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-45. Define MCU_E_ISR_INVALID_MODE_CONFIG Description

Name	MCU_E_ISR_INVALID_MODE_CONFIG
Initializer	((uint8)0x05U)

3.8.1.42 Define MCU_E_ISR_VOLTAGE_ERROR

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-46. Define MCU_E_ISR_VOLTAGE_ERROR Description

Name	MCU_E_ISR_VOLTAGE_ERROR
Initializer	((uint8)0x06U)

3.8.1.43 Define MCU_E_ISR_TEMPERATURE_MONITOR_ERROR

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-47. Define MCU_E_ISR_TEMPERATURE_MONITOR_ERROR Description

Name	MCU_E_ISR_TEMPERATURE_MONITOR_ERROR
Initializer	((uint8)0x07U)

3.8.1.44 Define MCU_E_ISR_FUNC_RESET_ALT_FAILURE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-48. Define MCU_E_ISR_FUNC_RESET_ALT_FAILURE Description

Name	MCU_E_ISR_FUNC_RESET_ALT_FAILURE
Initializer	((uint8)0x08U)

3.8.1.45 Define MCU_E_ISR_DEST_RESET_ALT_FAILURE

Error ISR values are of type uint8. The following error codes are reported by the error ISR.

Table 3-49. Define MCU_E_ISR_DEST_RESET_ALT_FAILURE Description

Name	MCU_E_ISR_DEST_RESET_ALT_FAILURE
Initializer	((uint8)0x09U)

3.8.1.46 Define MCU_CMU_CLEAR_CLOCK_IRQ_FLAG

Table 3-50. Define MCU_CMU_CLEAR_CLOCK_IRQ_FLAG Description

Name	MCU_CMU_CLEAR_CLOCK_IRQ_FLAG
Initializer	(STD_OFF)

3.8.1.47 Define MCU_DEV_ERROR_DETECT

(MCU166)Pre-processor switch for enabling the development error detection and reporting to the DET. (MCU100) The detection of development errors is configurable (ON / OFF) at pre-compile time.

Satisfied Requirements: MCU166, MCU100

Table 3-51. Define MCU_DEV_ERROR_DETECT Description

Name	MCU_DEV_ERROR_DETECT
Initializer	(STD_ON)

3.8.1.48 Define MCU_VERSION_INFO_API

(MCU168)Pre-processor switch to enable/disable the API to read out the modules version information.

Satisfied Requirements: MCU168

Table 3-52. Define MCU_VERSION_INFO_API Description

Name	MCU_VERSION_INFO_API
Initializer	(STD_ON)

3.8.1.49 Define MCU_GET_RAM_STATE_API

(MCU181)Pre-processor switch to enable/disable the API Mcu_GetRamState.

Satisfied Requirements: MCU181

Table 3-53. Define MCU_GET_RAM_STATE_API Description

Name	MCU_GET_RAM_STATE_API
Initializer	(STD_ON)

3.8.1.50 Define MCU_INIT_CLOCK

(MCU182)If this parameter is set to FALSE, the clock initialization has to be disabled from the MCU driver.

Satisfied Requirements: MCU182

Table 3-54. Define MCU_INIT_CLOCK Description

Name	MCU_INIT_CLOCK
Initializer	(STD_ON)

3.8.1.51 Define MCU_NO_PLL

(MCU180)This parameter shall be set True, if the H/W does not have a PLL or the PLL circuitry is enabled after the power on without S/W intervention.

Satisfied Requirements: MCU180

Table 3-55. Define MCU_NO_PLL Description

Name	MCU_NO_PLL
Initializer	(STD_OFF)

3.8.1.52 Define MCU_ENTER_LOW_POWER_MODE

Support for Low Power mode. If this parameter has been configured to 'TRUE', the function 'Mcu_SetMode()' shall not be impacted and behave as specified.

Satisfied Requirements: PR-MCAL-3184

Table 3-56. Define MCU_ENTER_LOW_POWER_MODE Description

Name	MCU_ENTER_LOW_POWER_MODE
Initializer	(STD_ON)

3.8.1.53 Define MCU_PERFORM_RESET_API

(MCU146)The function Mcu_PerformReset is only available if the runtime parameter McuPerformResetApi is set to TRUE.

Satisfied Requirements: MCU146

Table 3-57. Define MCU_PERFORM_RESET_API Description

Name	MCU_PERFORM_RESET_API
Initializer	(STD_ON)

3.8.1.54 Define MCU_TIMEOUT_LOOPS

Timeout representing the number of loops for preventing to lock inside am infinite while/for.

Table 3-58. Define MCU_TIMEOUT_LOOPS Description

Name	MCU_TIMEOUT_LOOPS
Initializer	(uint32)10000U

3.8.1.55 Define MCU_RESET_CALLOUT

The callout reset procedure configured by the user.

Table 3-59. Define MCU_RESET_CALLOUT Description

Name	MCU_RESET_CALLOUT
Initializer	The name of the function as defined in the Mcu configuration

3.8.1.56 Define MCU_RESET_CALLOUT_USED

The user callout reset is/isn't available (STD_ON/STD_OFF) - called by MCU right before`Mcu_PerformReset()`.

Table 3-60. Define MCU_RESET_CALLOUT_USED Description

Name	MCU_RESET_CALLOUT_USED
Initializer	(STD_OFF)

3.8.1.57 Define MCU_CMU_FCCU_NOTIFICATION

The callout configured by the user for CMU notifications.

Table 3-61. Define MCU_CMU_FCCU_NOTIFICATION Description

Name	MCU_CMU_FCCU_NOTIFICATION
Initializer	FUNC(void, MCU_CODE) MCU_CMU_FCCU_NOTIFICATION(VAR (uint8, AUTOMATIC) u8IndexCmu)

3.8.1.58 Define MCU_ERROR_ISR_NOTIFICATION

The callout configured by the user for CMU notifications.

Table 3-62. Define MCU_ERROR_ISR_NOTIFICATION Description

Name	MCU_ERROR_ISR_NOTIFICATION
Initializer	FUNC(void, MCU_CODE) MCU_ERROR_ISR_NOTIFICATION(VAR (uint8, AUTOMATIC) u8ErrorCode)

3.8.1.59 Define MCU_TRANSITION_COMPLETE_ISR_USED

ISR Mcu_ModeTransitionComplete_ISR complete is/isn't available (STD_ON/ STD_OFF). If at least one McuModuleConfiguration has the McuTransitionComplete parameter set true, then this precompile define is set to STD_ON.

Table 3-63. Define MCU_TRANSITION_COMPLETE_ISR_USED Description

Name	MCU_TRANSITION_COMPLETE_ISR_USED
Initializer	(STD_ON)

3.8.1.60 Define MCU_SAFE_MODE_ISR_USED

ISR Mcu_ModeEnterSafe_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-64. Define MCU_SAFE_MODE_ISR_USED Description

Name	MCU_SAFE_MODE_ISR_USED
Initializer	(STD_ON)

3.8.1.61 Define MCU_RGM_CLEAR_CLOCK_IRQ_FLAG

Table 3-65. Define MCU_RGM_CLEAR_CLOCK_IRQ_FLAG Description

Name	MCU_RGM_CLEAR_CLOCK_IRQ_FLAG
Initializer	(STD_OFF)

3.8.1.62 Define MCU_INVALID_MODE_ISR_USED

ISR Mcu_ModeInvalid_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-66. Define MCU_INVALID_MODE_ISR_USED Description

Name	MCU_INVALID_MODE_ISR_USED
Initializer	(STD_ON)

3.8.1.63 Define MCU_INVALID_CONFIGURATION_ISR_USED

ISR Mcu_ModeInvalidConfig_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-67. Define MCU_INVALID_CONFIGURATION_ISR_USED Description

Name	MCU_INVALID_CONFIGURATION_ISR_USED
Initializer	(STD_ON)

3.8.1.64 Define MCU_INVALID_CLK_CONFIGURATION_ISR_USED

ISR Mcu_ModeInvalidConfig_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-68. Define MCU_INVALID_CLK_CONFIGURATION_ISR_USED Description

Name	MCU_INVALID_CLK_CONFIGURATION_ISR_USED
Initializer	(STD_OFF)

3.8.1.65 Define MCU_INVALID_CORE_CONFIGURATION_ISR_USED

ISR MCU_INVALID_CORE_CONFIGURATION_ISR_USED is/isn't available (STD_ON/STD_OFF).

Table 3-69. Define MCU_INVALID_CORE_CONFIGURATION_ISR_USED Description

Name	MCU_INVALID_CORE_CONFIGURATION_ISR_USED
Initializer	(STD_OFF)

3.8.1.66 Define MCU_RESET_ALTERNATE_ISR_USED

ISR Mcu_ResetAlternate_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-70. Define MCU_RESET_ALTERNATE_ISR_USED Description

Name	MCU_RESET_ALTERNATE_ISR_USED
Initializer	(STD_OFF)

3.8.1.67 Define MCU_VOLTAGE_ERROR_ISR_USED

ISR Mcu_VoltageError_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-71. Define MCU_VOLTAGE_ERROR_ISR_USED Description

Name	MCU_VOLTAGE_ERROR_ISR_USED
Initializer	(STD_OFF)

3.8.1.68 Define MCU_TEMPERATURE_ERROR_ISR_USED

ISR Mcu_VoltageError_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-72. Define MCU_TEMPERATURE_ERROR_ISR_USED Description

Name	MCU_TEMPERATURE_ERROR_ISR_USED
Initializer	(STD_OFF)

3.8.1.69 Define MCU_CMU_ERROR_ISR_USED

ISR Mcu_CmuClockFail_ISR is always available.

Table 3-73. Define MCU_CMU_ERROR_ISR_USED Description

Name	MCU_CMU_ERROR_ISR_USED
Initializer	(STD_ON)

3.8.1.70 Define MCU_GET_PERIPH_STATE_API

Enable the usage of Non-Autosar API Mcu_GetPeripheral_State() for getting infos about peripheral state from MC_ME module.

Table 3-74. Define MCU_GET_PERIPH_STATE_API Description

Name	MCU_GET_PERIPH_STATE_API
Initializer	(STD_OFF)

3.8.1.71 Define MCU_GET_SYSTEM_STATE_API

Enable the usage of Non-Autosar API Mcu_GetSystem_State() for getting infos system platform configuration.

Table 3-75. Define MCU_GET_SYSTEM_STATE_API Description

Name	MCU_GET_SYSTEM_STATE_API
Initializer	(STD_OFF)

3.8.1.72 Define MCU_POWERMODE_STATE_API

Enable the usage of Non-Autosar API Mcu_GetPowerMode_State() for getting infos system platform configuration.

Table 3-76. Define MCU_POWERMODE_STATE_API Description

Name	MCU_POWERMODE_STATE_API
Initializer	(STD_OFF)

3.8.1.73 Define MCU_GET_POWER_DOMAIN_API

Enable the usage of Non-Autosar API Mcu_GetPowerDomain_Status. Get PMU state: operable or not.

Table 3-77. Define MCU_GET_POWER_DOMAIN_API Description

Name	MCU_GET_POWER_DOMAIN_API
Initializer	(STD_ON)

3.8.1.74 Define MCU_PLLDIG_PLL0_EXTPD_ISR_USED

ISR Mcu_PllDigPll0_PowerDownComplete_ISR is/isn't available (STD_ON/STD_OFF)

Table 3-78. Define MCU_PLLDIG_PLL0_EXTPD_ISR_USED Description

Name	MCU_PLLDIG_PLL0_EXTPD_ISR_USED
Initializer	(STD_OFF)

3.8.1.75 Define MCU_PLLDIG_PLL0_LOL_ISR_USED

ISR Mcu_PllDigPll0_LossOfLock_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-79. Define MCU_PLLDIG_PLL0_LOL_ISR_USED Description

Name	MCU_PLLDIG_PLL0_LOL_ISR_USED
Initializer	(STD_OFF)

3.8.1.76 Define MCU_SXOSC_INT_USED

ISR Mcu_AfeSDPllLock_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-80. Define MCU_SXOSC_INT_USED Description

Name	MCU_SXOSC_INT_USED
Initializer	(STD_OFF)

3.8.1.77 Define MCU_FXOSC_INT_USED

ISR Mcu_AfeSDPllLock_ISR is/isn't available (STD_ON/STD_OFF).

Table 3-81. Define MCU_FXOSC_INT_USED Description

Name	MCU_FXOSC_INT_USED
Initializer	(STD_OFF)

3.8.1.78 Define MCU_GET_MEM_CONFIG_API

Enable the usage of Non-Autosar API`Mcu_SscmGetMemConfig()`. Get SSCM_MEMCONFIG.

Table 3-82. Define MCU_GET_MEM_CONFIG_API Description

Name	MCU_GET_MEM_CONFIG_API
Initializer	(STD_ON)

3.8.1.79 Define MCU_SSCM_GET_STATUS_API

Enable the usage of Non-Autosar API Mcu_SscmGetStatus(). Get SSCM_STATUS.

Table 3-83. Define MCU_SSCM_GET_STATUS_API Description

Name	MCU_SSCM_GET_STATUS_API
Initializer	(STD_OFF)

3.8.1.80 Define MCU_SSCM_GET_UOPS_API

Enable the usage of Non-Autosar API Mcu_SscmGetUops(). Get SSCM_UOPS.

Table 3-84. Define MCU_SSCM_GET_UOPS_API Description

Name	MCU_SSCM_GET_UOPS_API
Initializer	(STD_OFF)

3.8.1.81 Define MCU_GET_MIDR_API

Enable the usage of Non-Autosar API`Mcu_GetMidrStructure()`. Returns .

Table 3-85. Define MCU_GET_MIDR_API Description

Name	MCU_GET_MIDR_API
Initializer	(STD_ON)

3.8.1.82 Define MCU_EMIOS_CONFIGURE_GPREN_API

Enable the usage of Non-Autosar API Mcu_EmiosConfigureGpren().

Table 3-86. Define MCU_EMIOS_CONFIGURE_GPREN_API Description

Name	MCU_EMIOS_CONFIGURE_GPREN_API
Initializer	(STD_OFF)

3.8.1.83 Define MCU_DISABLE_CMU_API

Enable the usage of Non-Autosar API Mcu_Disable_CMU().

Table 3-87. Define MCU_DISABLE_CMU_API Description

Name	MCU_DISABLE_CMU_API
Initializer	(STD_OFF)

3.8.1.84 Define MCU_DISABLE_DEM_REPORT_ERROR_STATUS

Enable/Disable the API for reporting the Dem Error.

Table 3-88. Define MCU_DISABLE_DEM_REPORT_ERROR_STATUS Description

Name	MCU_DISABLE_DEM_REPORT_ERROR_STATUS
Initializer	(STD_OFF)

3.8.1.85 Define MCU_CONFIGURE_CADDRN

Enable writing the MCU_MC_ME_CADDRn registers.

Table 3-89. Define MCU_CONFIGURE_CADDRN Description

Name	MCU_CONFIGURE_CADDRN
Initializer	(STD_ON)

3.8.1.86 Define MCU_FAST_MODE_CONFIG

This define controls the availability of the Mcu_MC_ME_FastModeConfig function.

Table 3-90. Define MCU_FAST_MODE_CONFIG Description

Name	MCU_FAST_MODE_CONFIG
Initializer	(STD_OFF)

3.8.1.87 Define MCU_LPU_SUPPORT

This define controls the availability of the Low Power Subsystem.

Table 3-91. Define MCU_LPU_SUPPORT Description

Name	MCU_LPU_SUPPORT
Initializer	(STD_OFF)

3.8.1.88 Define MCU_MAX_CLKCONFIGS

Table 3-92. Define MCU_MAX_CLKCONFIGS Description

Name	MCU_MAX_CLKCONFIGS
Initializer	((uint32)[!"num:i($MaxNoOfClkCfgs)"!]U)

3.8.1.89 Define MCU_MAX_MODECONFIGS

Table 3-93. Define MCU_MAX_MODECONFIGS Description

Name	MCU_MAX_MODECONFIGS
Initializer	((uint32)[!"num:i($MaxNoOfModeCfgs)"!]U)

3.8.1.90 Define MCU_MAX_RAMCONFIGS

Maximum number of MCU Clock configurations.

Table 3-94. Define MCU_MAX_RAMCONFIGS Description

Name	MCU_MAX_RAMCONFIGS
Initializer	((uint32)[!"num:i($MaxNoOfRamCfgs)"!]U)

3.8.1.91 Define MCU_PRECOMPILE_SUPPORT

Pre-compile Support.

Table 3-95. Define MCU_PRECOMPILE_SUPPORT Description

Name	MCU_PRECOMPILE_SUPPORT
Initializer	[!IF "(IMPLEMENTATION_CONFIG_VARIANT = 'VariantPreCompile') and (variant:size() <= 1)"!](STD_ON)[!ELSE!](STD_OFF)[!ENDIF!]

3.8.1.92 Define MCU_CMU_UNITS

Number of available CMU units.

Table 3-96. Define MCU_CMU_UNITS Description

Name	MCU_CMU_UNITS
Initializer	(uint8)13U

3.8.1.93 Define MCU_CGM_NUMBER_OF_SYSTEM_CLOCK_REGS

max number of system clock regs

Table 3-97. Define MCU_CGM_NUMBER_OF_SYSTEM_CLOCK_REGS Description

Name	MCU_CGM_NUMBER_OF_SYSTEM_CLOCK_REGS
Initializer	((uint8)6U)

3.8.1.94 Define MCU_MC_ME_NUMBER_OF_PCTL_REGS

number of PLL regs used on the current platform

Table 3-98. Define MCU_MC_ME_NUMBER_OF_PCTL_REGS Description

Name	MCU_MC_ME_NUMBER_OF_PCTL_REGS
Initializer	((uint8)67U)

3.8.1.95 Define MCU_CGM_NUMBER_OF_PLL_REGS

number of PLL regs used on the current platform

Table 3-99. Define MCU_CGM_NUMBER_OF_PLL_REGS Description

Name	MCU_CGM_NUMBER_OF_PLL_REGS
Initializer	((uint8)6U)

3.8.1.96 Define MCU_CGM_NUMBER_OF_AUX_CLK_REGS

number of PLL regs used on the current platform

Table 3-100. Define MCU_CGM_NUMBER_OF_AUX_CLK_REGS Description

Name	MCU_CGM_NUMBER_OF_AUX_CLK_REGS
Initializer	((uint8)10U)

3.8.1.97 Define MCU_EMIOS_SUPPORT

The eMIOS is configured or not.

Table 3-101. Define MCU_EMIOS_SUPPORT Description

Name	MCU_EMIOS_SUPPORT
Initializer	(STD_OFF)

3.8.1.98 Define MCU_CHECK_EMIOS_STATUS

Disable the usage of API Mcu_IPW_CheckEmiosStatus() for checking the eMIOS module before accessing to it.

Table 3-102. Define MCU_CHECK_EMIOS_STATUS Description

Name	MCU_CHECK_EMIOS_STATUS
Initializer	(STD_OFF)

3.8.1.99 Define MCU_EMIOS_NB_MODULES

Number of available eMIOS units.

Table 3-103. Define MCU_EMIOS_NB_MODULES Description

Name	MCU_EMIOS_NB_MODULES
Initializer	((uint8)(3U)

3.8.1.100 Define MCU_PRAM_MULTIPLE_PRAM

Multiple PRAM is available or not.

Table 3-104. Define MCU_PRAM_MULTIPLE_PRAM Description

Name	MCU_PRAM_MULTIPLE_PRAM
Initializer	(STD_OFF)

3.8.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR MCU Driver software specification Version 4.2 Rev0002 .

3.8.2.1 Enumeration Mcu_StatusType

The MCU module's implementer shall avoid the integration of incompatible files.

Details:

This enumerated type contains the Mcu driver's possible states.

Table 3-105. Enumeration Mcu_StatusType Values

Name	Initializer	Description
MCU_UNINIT	0x3U	The Mcu driver is not uninitialized.
MCU_IDLE	0xCU	= 0xE1 The Mcu driver is currently idle.
MCU_BUSY	0xAU	= 0xD2 The Mcu driver is currently busy.

3.8.2.2 Enumeration Mcu_MemoryConfigStageType

The stage of the flash and ram controllers configuration.

Details:

This is used to specify the entry and exit point of the flash and ram controllers configuration.

Table 3-106. Enumeration Mcu_MemoryConfigStageType Values

Name	Initializer	Description
MCU_MEMORY_CONFIG_ENTRY_POINT	0x55U	
MCU_MEMORY_CONFIG_EXIT_POINT	0xAAU	

3.8.2.3 Enumeration Mcu_PllStatusType

Type of the return value of the function Mcu_GetPllStatus.

Details:

The type of Mcu_PllStatusType is an enumeration with the following values: MCU_PLL_LOCKED, MCU_PLL_UNLOCKED, MCU_PLL_STATUS_UNDEFINED.

Implements: Mcu_PllStatusType_enumeration

Table 3-107. Enumeration Mcu_PllStatusType Values

Name	Initializer	Description
MCU_PLL_LOCKED	0x33U	PLL is locked.
MCU_PLL_UNLOCKED	0xCCU	PLL is unlocked.
MCU_PLL_STATUS_UNDEFINED	0x5AU	PLL Status is unknown.

3.8.2.4 Enumeration Mcu_RamStateType

Ram State of the microcontroller.

Details:

This is the Ram State data type returned by the function`Mcu_GetRamState()`of the Mcu module.

Table 3-108. Enumeration Mcu_RamStateType Values

Name	Initializer	Description
MCU_RAMSTATE_INVALID	0U	RAM content is not valid or unknown (default).
MCU_RAMSTATE_VALID		RAM content is valid.

3.8.2.5 Enumeration Mcu_ResetType

The type Mcu_ResetType, represents the different reset that a specified MCU can have.

Details:

The MCU module shall provide at least the values MCU_POWER_ON_RESET and MCU_RESET_UNDEFINED for the enumeration Mcu_ResetType.

Implements: Mcu_ResetType_enumeration

Table 3-109. Enumeration Mcu_ResetType Values

Name	Initializer	Description
MCU_POWER_ON_RESET	0x00U	Power on event. RGM_DES[F_POR].
MCU_HSM_DEST_RESET		HSM Destructive Reset RGM_DES[F_HSM_DEST].
MCU_SOFT_DEST_RESET		Software destructive reset. RGM_DES[F_SOFT_DESC].
MCU_SSCM_SEC_RESET		SSCM Secure reset (SSSR) RGM_DES[F_SSCM_SEC].
MCU_FUNC_ESC_RESET		Functional reset escalation. RGM_DES[F_FUNC_ESC].
MCU_SUF_RESET		SUF (STCU unrecoverable fault). RGM_DES[F_SUF].
MCU_SWT0_RESET		SWT0 (Flash Initialization Failure) Reset. RGM_DES[F_SWT0_RES].
MCU_SWT1_RESET		SWT1 (Flash Initialization Failure) Reset. RGM_DES[F_SWT1_RES].
MCU_SWT2_RESET		SWT2 (Flash Initialization Failure) Reset. RGM_DES[F_SWT2_RES].
MCU_EXR_RESET		External reset event. RGM_FES[F_EXR].
MCU_HSM_FUNC_RESET		HSM Generated Functional Reset. RGM_FES[F_HSM_FUNC].
MCU_SOFT_FUNC_RESET		Software destrutive event. RGM_FES[F_SOFT_FUNC].
MCU_NMI_WKPU_RESET		Non Maskable Interrupt from Wakeup Unit. RGM_FES[F_NMI_WKPU].

Table continues on the next page...

Table 3-109. Enumeration Mcu_ResetType Values (continued)

Name	Initializer	Description
MCU_JTAG_FUNC_RESET		JTAG Functional reset event. RGM_FES[F_JTAG_FUNC].
MCU_ST_DONE_RESET		Self-test completed event. RGM_FES[F_ST_DONE].
MCU_CMU_OLR_RESET		OSC Frequency less than RC. RGM_FES[F_CMU_OLR].
MCU_FCCU_LONG_RESET		FCCU Long Functional Reset. RGM_FES[F_FCCU_LONG].
MCU_FCCU_SHORT_RESET		FCCU Short Functional Reset. RGM_FES[F_FCCU_SHORT].
MCU_Z4A_DBG_RESET		Z4A Debug Reset. RGM_FES[F_Z4A_DBG].
MCU_Z4B_DBG_RESET		Z4B Debug Reset. RGM_FES[F_Z4B_DBG].
MCU_Z2_DBG_RESET		Z2 Debug Reset. RGM_FES[F_Z2_DBG].
MCU_LVD_IO_A_HI_RESET		LVD IO A HI reset. RGM_FES[F_LVD_IO_A_HI].
MCU_HVD_LV_COLD_RESET		High Voltage Detect. RGM_FES[F_HVD_LV_cold].
MCU_LVD_LV_PD2_COLD_RESET		LVD LV PD2 COLD reset. RGM_FES[F_LVD_LV_PD2_COLD].
MCU_NO_RESET_REASON		No reset reason found.
MCU_MULTIPLE_RESET_REASON		More than one reset events are logged except "Power on event".
MCU_RESET_UNDEFINED		Undefined reset source.

3.8.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR MCU Driver software specification Version 4.2 Rev0002 .

3.8.3.1 Function Mcu_Init

MCU driver initialization function.

Details:

This routine initializes the MCU Driver. The intention of this function is to make the configuration setting for power down, clock and Ram sections visible within the MCU Driver.

Return: void.

Implements: Mcu_Init_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Violates: MISRA 2004 Required Rule 19.15, Repeated include file

Prototype: `void Mcu_Init(const Mcu_ConfigType *ConfigPtr);`

Table 3-110. Mcu_Init Arguments

Type	Name	Direction	Description
const`Mcu_ConfigType`*	ConfigPtr	**input**	Pointer to configuration structure.

3.8.3.2 Function Mcu_InitRamSection

MCU driver initialization of Ram sections.

Details:

Function intializes the ram section selected by RamSection parameter. The section base address, size and value to be written are provided from the configuration structure. The function will write the value specified in the configuration structure indexed by RamSection. After the write it will read back the RAM to verify that the requested value was written.

Return: Command has or has not been accepted.

Implements: Mcu_InitRamSection_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Std_ReturnType Mcu_InitRamSection(Mcu_RamSectionType RamSection);`

Table 3-111. Mcu_InitRamSection Arguments

Type	Name	Direction	Description
`Mcu_RamSectionType`	RamSection	**input**	Index of ram section from config structure to be initialized.

Table 3-112. Mcu_InitRamSection Return Values

Name	Description
E_OK	Valid parameter, the driver state allowed execution and the RAM check was successful.
E_NOT_OK	Invalid parameter, the driver state did not allowed execution or the RAM check was not successful.

3.8.3.3 Function Mcu_InitClock

MCU driver clock initialization function.

Details:

This function intializes the PLL and MCU specific clock options. The clock setting is provided from the configuration structure.

Return: Command has or has not been accepted.

Implements: Mcu_InitClock_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Std_ReturnType Mcu_InitClock(Mcu_ClockType ClockSetting);`

Table 3-113. Mcu_InitClock Arguments

Type	Name	Direction	Description
`Mcu_ClockType`	ClockSetting	**input**	Clock setting ID from config structure to be used.

Table 3-114. Mcu_InitClock Return Values

Name	Description
E_OK	The driver state allowed the execution of the function and the provided parameter was in range.
E_NOT_OK	The driver state did not allowed execution or the parameter was invalid.

3.8.3.4 Function Mcu_SetMode

This function sets the MCU power mode.

Details:

This function activates MCU power mode from config structure selected by McuMode parameter. If the driver state is invalid or McuMode is not in range the function will skip changing the mcu mode.

Return: void.

Implements: Mcu_SetMode_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `void Mcu_SetMode(Mcu_ModeType McuMode);`

Table 3-115. Mcu_SetMode Arguments

Type	Name	Direction	Description
`Mcu_ModeType`	McuMode	**input**	MCU mode setting ID from config structure to be set.

3.8.3.5 Function Mcu_DistributePllClock

This function activates the PLL clock to the MCU clock distribution.

Details:

Function completes the PLL configuration and then activates the PLL clock to MCU. If the MCU_NO_PLL is TRUE the Mcu_DistributePllClock has to be disabled. The function will not distribute the PLL clock if the driver state does not allow it, or the PLL is not stable.

Return: Std_ReturnType.

Implements: Mcu_DistributePllClock_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Std_ReturnType Mcu_DistributePllClock( void);`

3.8.3.6 Function Mcu_GetPllStatus

This function returns the lock status of the PLL.

Details:

The user takes care that the PLL is locked by executing Mcu_GetPllStatus. If the MCU_NO_PLL is TRUE the MCU_GetPllStatus has to return MCU_PLL_STATUS_UNDEFINED. It will also return MCU_PLL_STATUS_UNDEFINED if the driver state was invalid

Return: Provides the lock status of the PLL.

Implements: Mcu_GetPllStatus_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Mcu_PllStatusType Mcu_GetPllStatus(void);`

Table 3-116. Mcu_GetPllStatus Return Values

Name	Description
MCU_PLL_STATUS_UNDEFINED	PLL Status is unknown.
MCU_PLL_LOCKED	PLL is locked.
MCU_PLL_UNLOCKED	PLL is unlocked.

3.8.3.7 Function Mcu_GetResetReason

This function returns the Reset reason.

Details:

This routine returns the Reset reason that is read from the hardware.

Return: Reason of the Reset event.

Implements: Mcu_GetResetReason_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Mcu_ResetType Mcu_GetResetReason(void);`

Table 3-117. Mcu_GetResetReason Return Values

Name	Description
MCU_POR_RESET	Power on event.
MCU_SOFT_DEST_RESET	Software destructive reset.
MCU_FFRR_RESET	FCCU failure to react reset.

Table continues on the next page...

Table 3-117. Mcu_GetResetReason Return Values (continued)

Name	Description
MCU_EDR_RESET	Functional reset escalation.
MCU_TSR_DEST_RESET	Temperature sensor destructive reset.
MCU_VOR_DEST_RESET	Voltage out of range destructive reset.
MCU_EXR_RESET	External reset event.
MCU_ST_DONE_RESET	Self-test completed event.
MCU_SOFT_FUNC_RESET	Software destrutive event.
MCU_FCCU_HARD_RESET	FCCU hard reaction request event.
MCU_FCCU_SOFT_RESET	FCCU soft reaction request event.
MCU_JTAG_RESET	JTAG initiated reset event.
MCU_TSR_FUNC_RESET	Temperature sensor functional reset.
MCU_VOR_FUNC_RESET	Voltage out of range functional reset.

3.8.3.8 Function Mcu_GetResetRawValue

This function returns the Raw Reset value.

Details:

This routine returns the Raw Reset value that is read from the hardware.

Return: Description of the returned value.

Implements: Mcu_GetResetRawValue_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Mcu_RawResetType Mcu_GetResetRawValue(void);`

Table 3-118. Mcu_GetResetRawValue Return Values

Name	Description
uint32	Code of the Raw reset value. The bit order in the returned value is: [0] - MC_RGM_FES_F_EXR_MASK32 [1] - MC_RGM_FES_F_ST_DONE_MASK32 [2] - MC_RGM_FES_F_SOFT_FUNC_MASK32 [3] - MC_RGM_FES_F_FCCU_HARD_MASK32 [4] - MC_RGM_FES_F_FCCU_SOFT_MASK32 [5] - MC_RGM_FES_F_JTAG_FUNC_MASK32 [6] - MC_RGM_FES_F_TSR_FUNC_MASK32 [7] - MC_RGM_FES_F_VOR_FUNC_MASK32 [16] - MC_RGM_DES_F_POR_MASK32 [17] - MC_RGM_DES_F_SOFT_DEST_MASK32 [18] - MC_RGM_DES_F_FFRR_MASK32 [19] - MC_RGM_DES_F_EDR_MASK32 [20] - MC_RGM_DES_F_TSR_DEST_MASK32 [21] - MC_RGM_DES_F_VOR_DEST_MASK32.

3.8.3.9 Function Mcu_PerformReset

This function performs a microcontroller reset.

Details:

This function performs a microcontroller reset by using the hardware feature of the microcontroller. In case the function returns, the user must reset the platform using an alternate reset mechanism

Return: void.

Implements: Mcu_PerformReset_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `void Mcu_PerformReset(void);`

3.8.3.10 Function Mcu_GetVersionInfo

This function returns the Version Information for the MCU module.

Details:

This function returns the vendor id, module id, major, minor and patch version.

Return: void.

Implements: Mcu_GetVersionInfo_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `void Mcu_GetVersionInfo(Std_VersionInfoType *versioninfo);`

Table 3-119. Mcu_GetVersionInfo Arguments

Type	Name	Direction	Description
Std_VersionInfoType *	versioninfo	**input, output**	A pointer to a variable to store version info.

3.8.3.11 Function Mcu_GetRamState

This function returns the actual state of the RAM.

Details:

This function returns if the Ram Status is valid after a reset. The report is get from STCU as a result of MBIST (Memory Built-In Self Tests).

Return: Status of the Ram Content.

Implements: Mcu_GetRamState_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `Mcu_RamStateType Mcu_GetRamState(void);`

Table 3-120. Mcu_GetRamState Return Values

Name	Description
MCU_RAMSTATE_INVALID	Ram state is not valid or unknown (default), or the driver state does not allow this call.
MCU_RAMSTATE_VALID	Ram state is valid.

3.8.3.12 Function Mcu_GetPeripheralState

Prototype: `Std_ReturnType Mcu_GetPeripheralState(Mcu_PeripheralId McuPeriphId);`

3.8.3.13 Function Mcu_GetSystemState

Prototype: `uint32 Mcu_GetSystemState(void);`

3.8.3.14 Function Mcu_GetPowerModeState

Prototype: `Mcu_PowerModeStateType Mcu_GetPowerModeState(void);`

3.8.3.15 Function Mcu_GetPowerDomainState

Prototype: `Std_ReturnType Mcu_GetPowerDomainState(void);`

3.8.3.16 Function Mcu_SscmGetMemConfig

Prototype: `Mcu_SSCM_MemConfigType Mcu_SscmGetMemConfig(void);`

3.8.3.17 Function Mcu_SscmGetStatus

Prototype: `Mcu_SSCM_StatusType Mcu_SscmGetStatus(void);`

3.8.3.18 Function Mcu_SscmGetUops

Prototype: `Mcu_SSCM_UopsType Mcu_SscmGetUops(void);`

3.8.3.19 Function Mcu_GetMidrStructure

This function returns the value of the MIDR registers.

Details:

This function returns the platform dependent Mcu_MidrReturnType structure witch contains the MIDRn registers.

Return: void.

Implements: Mcu_GetMidrStructure_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `void Mcu_GetMidrStructure(Mcu_MidrReturnType *midr);`

Table 3-121. Mcu_GetMidrStructure Arguments

Type	Name	Direction	Description
Mcu_MidrReturnType *	midr	**input, output**	A pointer to a variable to store the Mcu_MidrReturnType structure.

3.8.3.20 Function Mcu_Disable_CMU

Prototype: `void Mcu_Disable_CMU(uint8 u8IndexCmu);`

3.8.3.21 Function Mcu_EmiosConfigureGpren

eMios Global Prescaler Enable.

Details:

This function enables or disables the GPREN bit of the EMIOSMCR register of an addressed eMIOS instance.

Return: void.

Implements: Mcu_EmiosConfigureGpren_Activity

Violates: Violates MISRA 2004 Required Rule 8.10, global declaration of function

Prototype: `void Mcu_EmiosConfigureGpren(uint8 u8Module, uint8 u8Value);`

Table 3-122. Mcu_EmiosConfigureGpren Arguments

Type	Name	Direction	Description
`uint8`	u8Module	**input**	MCU_EMIOS_MODULE_0 --> Select eMios 0 MCU_EMIOS_MODULE_1 --> Select eMios 1 MCU_EMIOS_MODULE_2 --> Select eMios 2
`uint8`	u8Value	**input**	MCU_EMIOS_GPREN_BIT_ENABLE --> Global Prescaler Enabled MCU_EMIOS_GPREN_BIT_DISABLE --> Global Prescaler Disabled

3.8.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR MCU Driver software specification Version 4.2 Rev0002 .

3.8.4.1 Structure Mcu_ConfigType

Initialization data for the MCU driver.

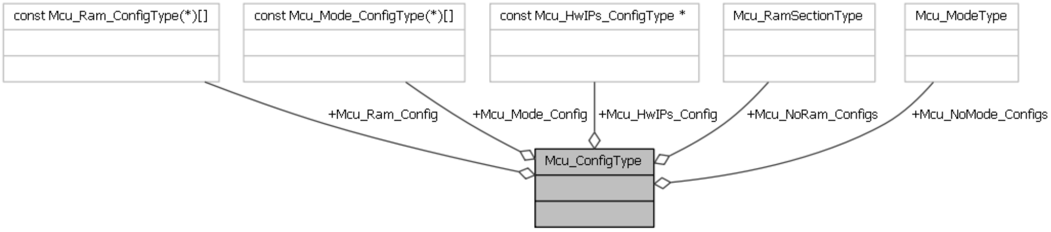


Figure 3-1. Struct Mcu_ConfigType

Details:

A pointer to such a structure is provided to the MCU initialization routines for configuration.

Implements: Mcu_ConfigType_struct

Declaration:

```
typedef struct
                           {
                             const Mcu_ModeConfigType(* Mcu_apModeConfig)[],
                                     constMcu_RamConfigType(* Mcu_apRamConfig)[],
                                     Mcu_ClockType Mcu_NoClkConfigs,
                                     Mcu_ModeType Mcu_NoModeConfigs,
                                     Mcu_RamSectionType Mcu_NoRamConfigs,
                                     constMcu_DemConfigType* Mcu_pDemConfig,
                                     constMcu_HwIPsConfigType* Mcu_pHwIPsConfig
                           } Mcu_ConfigType;
```

Table 3-123. Structure Mcu_ConfigType member description

Member	Description
Mcu_apModeConfig	
Mcu_apRamConfig	RAM data configuration.
Mcu_NoClkConfigs	
Mcu_NoModeConfigs	Total number of MCU clock configurations.
Mcu_NoRamConfigs	Total number of MCU modes.
Mcu_pDemConfig	DEM error reporting configuration.
Mcu_pHwIPsConfig	IPs data generic configuration.

3.8.4.2 Structure Mcu_RamConfigType

Definition of a RAM section within the configuration structure. The definitions for each RAM section within the structure`Mcu_ConfigType`shall contain:

- RAM section base address
- Section size
- Data pre-setting to be initialized.

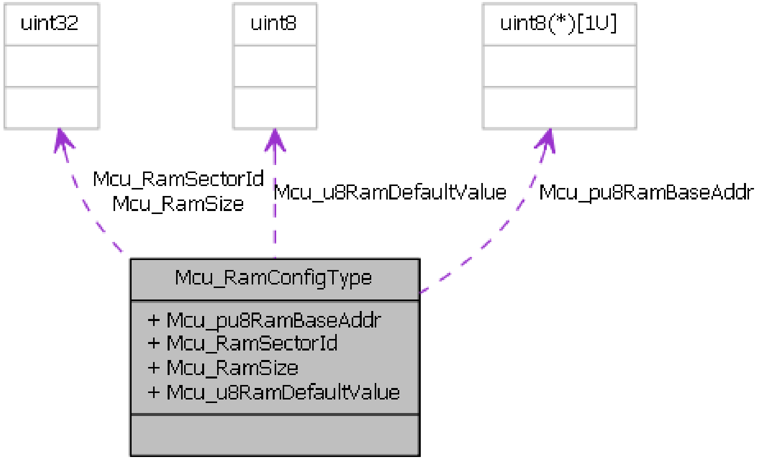


Figure 3-2. Struct Mcu_RamConfigType

Declaration:

```
typedef struct
                           {
                             uint8 * Mcu_pu8RamBaseAddr,
                                     Mcu_RamSectionType Mcu_RamSectorId,
                                     uint32 Mcu_u32RamSize,
                                     uint8 Mcu_u8RamDefaultValue
                           } Mcu_RamConfigType;
```

Table 3-124. Structure Mcu_RamConfigType member description

Member	Description
Mcu_pu8RamBaseAddr	RAM section base address.
Mcu_RamSectorId	The ID for Ram Sector configuration.
Mcu_u32RamSize	RAM section size.
Mcu_u8RamDefaultValue	RAM default value for initialization.

3.8.4.3 Structure Mcu_DemConfigType

DEM error reporting configuration.

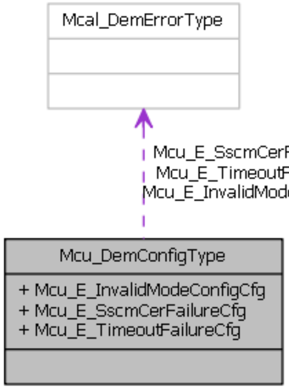

Mcal_DemErrorType
Mcu_E_SscmCerFailureCfg
Mcu_E_TimeoutFailureCfg
Mcu_E_InvalidModeConfigCfg
Mcu_DemConfigType
+ Mcu_E_InvalidModeConfigCfg
+ Mcu_E_SscmCerFailureCfg
+ Mcu_E_TimeoutFailureCfg

Figure 3-3. Struct Mcu_DemConfigType

Details:

This structure contains information DEM error reporting

Declaration:

```
typedef struct
                           {
                             Mcal_DemErrorType Mcu_E_SscmCerFailureCfg,
                                     Mcal_DemErrorType Mcu_E_TimeoutFailureCfg
                           } Mcu_DemConfigType;
```

Table 3-125. Structure Mcu_DemConfigType member description

Member	Description
Mcu_E_SscmCerFailureCfg	
Mcu_E_TimeoutFailureCfg	

3.8.5 Types Reference

Types supported by the driver are as per AUTOSAR MCU Driver software specification Version 4.2 Rev0002 .

3.8.5.1 Typedef Mcu_ClockType

Defines the identification (ID) for clock setting configured via the configuration structure.

Details:

The type shall be uint8, uint16 or uint32, depending on uC platform.

Implements: Mcu_ClockType_typedef

Type: uint32

3.8.5.2 Typedef Mcu_ModeType

The Mcu_ModeType specifies the identification (ID) for a MCU mode, configured via configuration structure.

Details:

The type shall be uint8, uint16 or uint32.

Implements: Mcu_ModeType_typedef

Type: uint32

3.8.5.3 Typedef Mcu_RamIndexType

The Mcu_RamIndexType specifies the variable for indexinf RAM sections. The type shall be uint8, uint16 or uint32, based on best performance.

Type: uint32

3.8.5.4 Typedef Mcu_RamSectionType

The Mcu_RamSectionType specifies the identification (ID) for a RAM section, configured via the configuration structure. The type shall be uint8, uint16 or uint32, based on best performance.

Implements: Mcu_RamSectionType_typedef

Type: uint32

3.8.5.5 Typedef Mcu_RamSizeType

The Mcu_RamSizeType specifies the RAM section size. The type shall be uint8, uint16 or uint32, based on best performance.

Type: uint32

3.8.5.6 Typedef Mcu_RawResetType

The type Mcu_RawResetType specifies the reset reason in raw register format, read from a reset status register.

Details:

The type shall be uint8, uint16 or uint32 based on best performance.

Implements: Mcu_RawResetType_typedefDestructive and Functional Reset Events Log.

Type: uint32

3.9 Symbolic Names Disclaimer

All containers having the symbolic name tag set as true in the Autosar schema will generate defines like:

#define <Container_Short_Name> <Container_ID>

For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments).

Chapter 4
Tresos Configuration Plug-in

This chapter describes the Tresos configuration plug-in for the MCU Driver. The most of the parameters are described below.

4.1 Configuration elements of Mcu

Included forms :

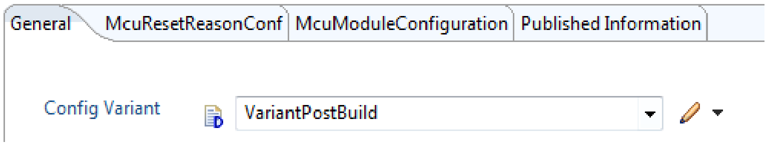
- IMPLEMENTATION_CONFIG_VARIANT
- McuGeneralConfiguration
- McuDebugConfiguration
- McuPublishedInformation
- CommonPublishedInformation
- McuModuleConfiguration

Table 4-1. Revision table

Revision	Date
0.8.0	2010-12-03

4.2 Form IMPLEMENTATION_CONFIG_VARIANT

VariantPreCompile: Only precompile time configuration parameters. Only one set of parameters. VariantPostBuild: Mix of precompile and postbuild time configuration parameters. Only one set of parameters. If Config Variant = VariantPreCompile, the files Port_Cfg.h and Port_Cfg.c should be used. If Config Variant = VariantPostBuild, the files Port_Cfg.h and Port_PBcfg.c should be used.



Config Variant: VariantPostBuild

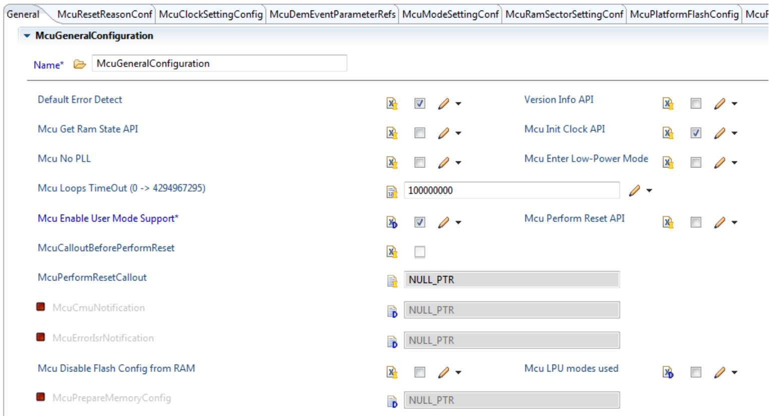
Figure 4-1. Tresos Plugin snapshot for IMPLEMENTATION_CONFIG_VARIANT form.

Table 4-2. Attribute IMPLEMENTATION_CONFIG_VARIANT detailed description

Property	Value
Label	Config Variant
Type	ENUMERATION
Default	VariantPostBuild
Range	VariantPostBuild VariantPreCompile

4.3 Form McuGeneralConfiguration

This container contains the general configuration for the MCU driver.



Name* McuGeneralConfiguration

Default Error Detect ☑
Version Info API □
Mcu Get Ram State API □
Mcu Init Clock API ☑
Mcu No PLL □
Mcu Enter Low-Power Mode □
Mcu Loops TimeOut (0 -> 4294967295) 100000000
Mcu Enable User Mode Support* ☑
Mcu Perform Reset API □
McuCalloutBeforePerformReset □
McuPerformResetCallout NULL_PTR
McuCmuNotification NULL_PTR
McuErrorIsrNotification NULL_PTR
Mcu Disable Flash Config from RAM □
Mcu LPU modes used □
McuPrepareMemoryConfig NULL_PTR

Figure 4-2. Tresos Plugin snapshot for McuGeneralConfiguration form.

4.3.1 McuDevErrorDetect (McuGeneralConfiguration)

Pre-processor switch for enabling the default error detection and reporting to the DET. The switch McuDevErrorDetect shall switch the Default Error Tracer (Det) detection and notification ON or OFF. The detection of default errors is configurable (ON/OFF) at precompile time. #define MCU_DEV_ERROR_DETECT (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-3. Attribute McuDevErrorDetect (McuGeneralConfiguration) detailed description

Property	Value
Label	Default Error Detect
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.3.2 McuVersionInfoApi (McuGeneralConfiguration)

Pre-processor switch to enable/disable the API to read out the modules version information. #define MCU_VERSION_INFO_API (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-4. Attribute McuVersionInfoApi (McuGeneralConfiguration) detailed description

Property	Value
Label	Version Info API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.3.3 McuGetRamStateApi (McuGeneralConfiguration)

Pre-processor switch to enable/disable the API Mcu_GetRamState. #define MCU_GET_RAM_STATE_API (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-5. Attribute McuGetRamStateApi (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Get Ram State API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.3.4 McuInitClock (McuGeneralConfiguration)

If this parameter is set to FALSE, the clock initialization has to be disabled from the MCU driver. This concept applies when there are some write once clock registers and a bootloader is present. If this parameter is set to TRUE, the MCU driver is responsible of the clock initialization #define MCU_INIT_CLOCK (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-6. Attribute McuInitClock (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Init Clock API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.5 McuNoPll (McuGeneralConfiguration)

This parameter shall be set True, if the H/W does not have a PLL or the PLL circuitry is enabled after the power on without S/W intervention. In this case MCU_DistributePllClock has to be disabled and MCU_GetPllStatus has to return MCU_PLL_STATUS_UNDEFINED. Otherwise this parameters has to be set False. #define MCU_NO_PLL (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-7. Attribute McuNoPll (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu No PLL
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	true

4.3.6 McuEnterLowPowerMode (McuGeneralConfiguration)

If this parameter has been configured to 'TRUE', the function 'Mcu_SetMode()' shall not be impacted and behave as specified. If this parameter has been configured to 'FALSE', the function 'Mcu_SetMode()' shall not perform the transition to any low power modes as are 'STOP' or 'STANDBY' or any other mode, where the core stops execution. #define MCU_ENTER_LOW_POWER_MODE (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-8. Attribute McuEnterLowPowerMode (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Enter Low-Power Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.3.7 McuTimeout (McuGeneralConfiguration)

This parameter represents the maximum number of loops for blocking functionality. The maximum time needed for a MC_ME transition from DRUN to DRUN with keeping PLL running is 3 ms. Please take this into consideration when choosing the value for this parameter.

Note: Implementation Specific Parameter.

Table 4-9. Attribute McuTimeout (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Loops TimeOut
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	50000
Invalid	Range >=0 <=4294967295

4.3.8 McuEnableUserModeSupport (McuGeneralConfiguration)

When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in

REG_PROT_GCR to 1. b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. For more information, please see chapter 5.7 User Mode Support in IM

Note: Implementation Specific Parameter.

Table 4-10. Attribute McuEnableUserModeSupport (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Enable User Mode Support
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.3.9 McuPerformResetApi (McuGeneralConfiguration)

Pre-processor switch to enable/disable the use the Mcu_PerformReset() API. OFF - Mcu_PerformReset() API is not used. ON - Mcu_PerformReset() API is used. #define MCU_PERFORM_RESET_API (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file.

Table 4-11. Attribute McuPerformResetApi (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Perform Reset API
Type	BOOLEAN
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	false

4.3.10 McuCalloutBeforePerformReset (McuGeneralConfiguration)

Check this if you want a callout function, called by MCU right before Mcu_PerformReset(). This parameter is available for configuration only if "McuPerformResetApi" is ON. #define MCU_RESET_CALLOUT_USED (STD_ON)/(STD_OFF) will be generated in Mcu_Cfg.h file. Note: Implementation Specific Parameter.

Table 4-12. Attribute McuCalloutBeforePerformReset (McuGeneralConfiguration) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.3.11 McuPerformResetCallout (McuGeneralConfiguration)

Function name of callout. The field is editable only if "McuCalloutBeforePerformReset" is ON. Note: Implementation Specific Parameter.

Table 4-13. Attribute McuPerformResetCallout (McuGeneralConfiguration) detailed description

Property	Value
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.3.12 McuCmuNotification (McuGeneralConfiguration)

Function pointer to callback function.

Table 4-14. Attribute McuCmuNotification (McuGeneralConfiguration) detailed description

Property	Value
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.3.13 McuErrorIsrNotification (McuGeneralConfiguration)

Function name of callout. This function will be called by the error ISR.

Note: Implementation Specific Parameter.

Table 4-15. Attribute McuErrorIsrNotification (McuGeneralConfiguration) detailed description

Property	Value
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.3.14 McuDisableFlashConfigFromRam (McuGeneralConfiguration)

Check this if you want the Flash configuration from RAM to be bypassed.

If this is checked the settings configured in McuPlatformFlashConfig and the settings from McuFlash (from McuClockSettingConfig) will not be used.

Note: Implementation Specific Parameter.

Table 4-16. Attribute McuDisableFlashConfigFromRam (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu Disable Flash Config from RAM
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.3.15 McuUseLpuModes (McuGeneralConfiguration)

Check this if you want to use LPU modes in the application.

If this is unchecked the code for LPU will be removed at pre-compile time.

Note: Implementation Specific Parameter.

Table 4-17. Attribute McuUseLpuModes (McuGeneralConfiguration) detailed description

Property	Value
Label	Mcu LPU modes used
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.3.16 McuPrepareMemoryConfig (McuGeneralConfiguration)

Function name of a callout that will be called before and after configuring the flash and ram controllers. It will have a parameter that will specify if it is the entry or the exit point of the controllers configuration.

Note: Implementation Specific Parameter.

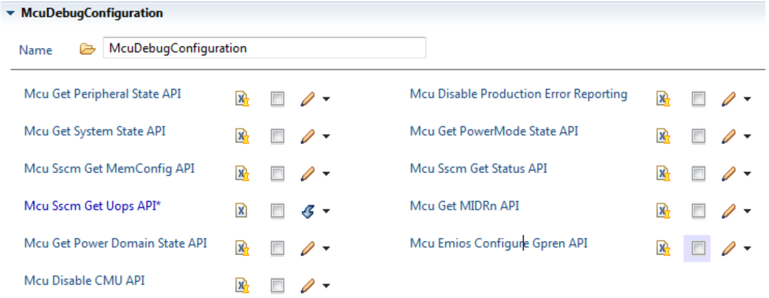
Table 4-18. Attribute McuPrepareMemoryConfig (McuGeneralConfiguration) detailed description

Property	Value
Type	FUNCTION-NAME
Origin	Custom
Symbolic Name	false
Default	NULL_PTR

4.4 Form McuDebugConfiguration

This container contains option for non-ASR APIs used for debug or extra-implementation. Note: Implementation Specific Parameter.

Form McuDebugConfiguration



McuDebugConfiguration

Name: McuDebugConfiguration

Mcu Get Peripheral State API	□	Mcu Disable Production Error Reporting	□
Mcu Get System State API	□	Mcu Get PowerMode State API	□
Mcu Sscm Get MemConfig API	□	Mcu Sscm Get Status API	□
Mcu Sscm Get Uops API*	□	Mcu Get MIDRn API	□
Mcu Get Power Domain State API	□	Mcu Emios Configure Gpren API	□
Mcu Disable CMU API	□		

Figure 4-3. Tresos Plugin snapshot for McuDebugConfiguration form.

4.4.1 McuDisableDemReportErrorStatus (McuDebugConfiguration)

Enable/Disable the API for reporting the Dem Error.

Note: Implementation Specific Parameter.

Table 4-19. Attribute McuDisableDemReportErrorStatus (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Disable Production Error Reporting
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.2 McuGetPeriphStateApi (McuDebugConfiguration)

Enable/Disable the API for checking peripheral states in the current mode from MC_ME configuration: Mcu_GetPeripheral_State(). E_OK means Peripheral with ID as parameter is clocked. E_NOT_OK means Peripheral with ID as parameter is not clocked. Note: Implementation Specific Parameter.

Table 4-20. Attribute McuGetPeriphStateApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Get Peripheral State API
Type	BOOLEAN
Origin	Custom

Table continues on the next page...

Table 4-20. Attribute McuGetPeriphStateApi (McuDebugConfiguration) detailed description (continued)

Property	Value
Symbolic Name	false
Default	false

4.4.3 McuGetSystemStatetApi (McuDebugConfiguration)

Enable/Disable the API for System state information: Mcu_GetSystem_State(). Information extracted from SSCM hw IP. Note: Implementation Specific Parameter.

Table 4-21. Attribute McuGetSystemStatetApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Get System State API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.4 McuGetPowerModeStatetApi (McuDebugConfiguration)

Enable/Disable the API for MC_ME state: Mcu_GetPowerMode_State(). Get information regarding current power mode, enabled clocks, etc (content of ME_GS register). Note: Implementation Specific Parameter.

Table 4-22. Attribute McuGetPowerModeStatetApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Get PowerMode State API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.5 McuSscmGetMemConfigApi (McuDebugConfiguration)

Enable/Disable the API for Mcu_SscmGetMemConfig().

Get information from SSCM_MEMCONFIG register.

Note: Implementation Specific Parameter.

Table 4-23. Attribute McuSscmGetMemConfigApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Sscm Get MemConfig API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.6 McuSscmGetStatusApi (McuDebugConfiguration)

Enable/Disable the API for Mcu_SscmGetStatus().

Get information from SSCM_STATUS register.

Note: Implementation Specific Parameter.

Table 4-24. Attribute McuSscmGetStatusApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Sscm Get Status API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.7 McuSscmGetUopsApi (McuDebugConfiguration)

Enable/Disable the API for Mcu_SscmGetUops().

Get information from SSCM_UOPS register.

Note: Implementation Specific Parameter.

Table 4-25. Attribute McuSscmGetUopsApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Sscm Get Uops API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.8 McuGetMidrStructureApi (McuDebugConfiguration)

Enable/Disable the API for Mcu_GetMidrStructure().

Get information from SIUL2 MIDRn registers.

Note: Implementation Specific Parameter.

Table 4-26. Attribute McuGetMidrStructureApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Get MIDRn API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.9 McuGetPowerDomainApi (McuDebugConfiguration)

Enable/Disable the API for MC_PCU state: Mcu_GetPowerDomain_Status(). Get information from PCU_STAT register. Note: Implementation Specific Parameter.

Table 4-27. Attribute McuGetPowerDomainApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Get Power Domain State API
Type	BOOLEAN

Table continues on the next page...

Table 4-27. Attribute McuGetPowerDomainApi (McuDebugConfiguration) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.4.10 McuEmiosConfigureGprenApi (McuDebugConfiguration)

Enable/Disable the API for Mcu_EmiosConfigureGpren().

Changes the GPREN bit of the EMIOSMCR register of an addressed eMIOS instance.

Note: Implementation Specific Parameter.

Table 4-28. Attribute McuEmiosConfigureGprenApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Emios Configure Gpren API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.4.11 McuDisableCmuApi (McuDebugConfiguration)

Enable/Disable the API for disabling the clock monitoring unit. Note: Implementation Specific Parameter.

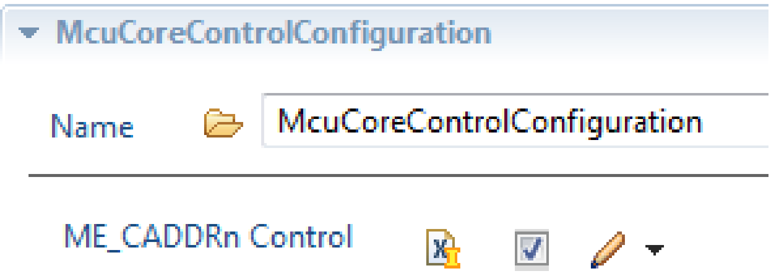
Table 4-29. Attribute McuDisableCmuApi (McuDebugConfiguration) detailed description

Property	Value
Label	Mcu Disable CMU API
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.5 Form McuCoreControlConfiguration

This configuration holds global control over the platform specific core control features.

This container is implementation specific.



McuCoreControlConfiguration

Name: McuCoreControlConfiguration

ME_CADDRn Control ☑

Figure 4-4. Tresos Plugin snapshot for McuCoreControlConfiguration form.

4.5.1 McuCADDRnControl (McuCoreControlConfiguration)

Global ENABLE / DISABLE of the code that writes the ME_CADDRn registers

From the user manual: This registers gives the boot address for z4a and a bit for controlling whether z4a is to be reset on the next mode change that has z4a configured to be running in the target mode.

This register is loaded from the flash during the boot process.

If this check box is ON, the registers will be written during each Mcu_SetMode() call

Note: Implementation Specific Parameter.

Table 4-30. Attribute McuCADDRnControl (McuCoreControlConfiguration) detailed description

Property	Value
Label	ME_CADDRn Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.6 Form McuEMIOSConfiguration

This configuration holds global control over the platform specific eMIOS control features.

This container is implementation specific.

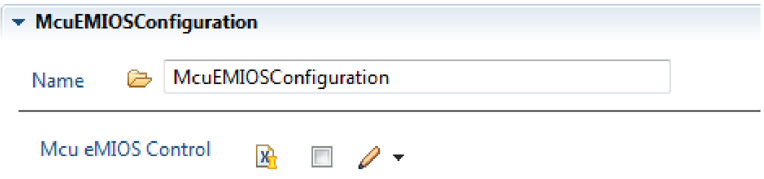


Figure 4-5. Tresos Plugin snapshot for McuEMIOSConfiguration form.

4.6.1 McuEMIOSControl (McuEMIOSConfiguration)

Global ENABLE / DISABLE of the code that writes the eMIOS200_MCR registers.

From the user manual: The MCR register contains Global Control bits for the eMIOS200 block.

If this check box is ON, the registers will be written during each Mcu_InitClock() call.

Note: Implementation Specific Parameter.

Table 4-31. Attribute McuEMIOSControl (McuEMIOSConfiguration) detailed description

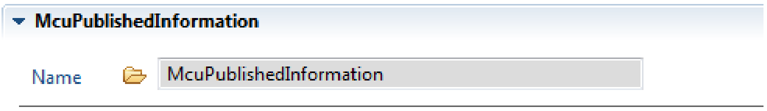
Property	Value
Label	Mcu eMIOS Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.7 Form McuPublishedInformation

Container holding all MCU specific published information parameters.

Included forms :

- Form McuResetReasonConf



McuPublishedInformation

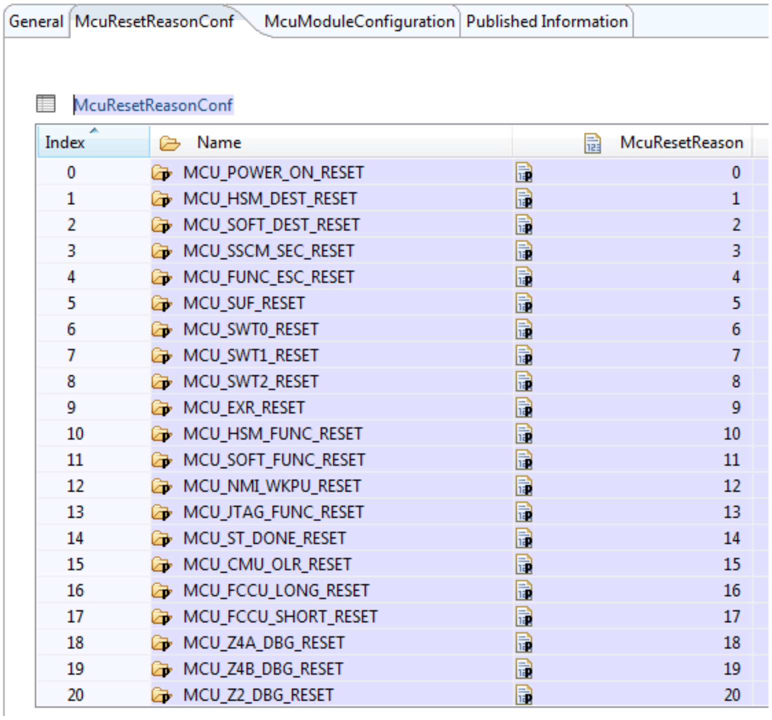
Name: McuPublishedInformation

Figure 4-6. Tresos Plugin snapshot for McuPublishedInformation form.

4.7.1 Form McuResetReasonConf

This container contains the configuration for the different type of reset reason that can be retrieved from Mcu_GetResetReason Api.

Is included by form : Form McuPublishedInformation



McuResetReasonConf

Index	Name	McuResetReason
0	MCU_POWER_ON_RESET	0
1	MCU_HSM_DEST_RESET	1
2	MCU_SOFT_DEST_RESET	2
3	MCU_SSCM_SEC_RESET	3
4	MCU_FUNC_ESC_RESET	4
5	MCU_SUF_RESET	5
6	MCU_SWT0_RESET	6
7	MCU_SWT1_RESET	7
8	MCU_SWT2_RESET	8
9	MCU_EXR_RESET	9
10	MCU_HSM_FUNC_RESET	10
11	MCU_SOFT_FUNC_RESET	11
12	MCU_NMI_WKPU_RESET	12
13	MCU_JTAG_FUNC_RESET	13
14	MCU_ST_DONE_RESET	14
15	MCU_CMU_OLR_RESET	15
16	MCU_FCCU_LONG_RESET	16
17	MCU_FCCU_SHORT_RESET	17
18	MCU_Z4A_DBG_RESET	18
19	MCU_Z4B_DBG_RESET	19
20	MCU_Z2_DBG_RESET	20

Figure 4-7. Tresos Plugin snapshot for McuResetReasonConf form.

4.7.1.1 McuResetReason (McuResetReasonConf)

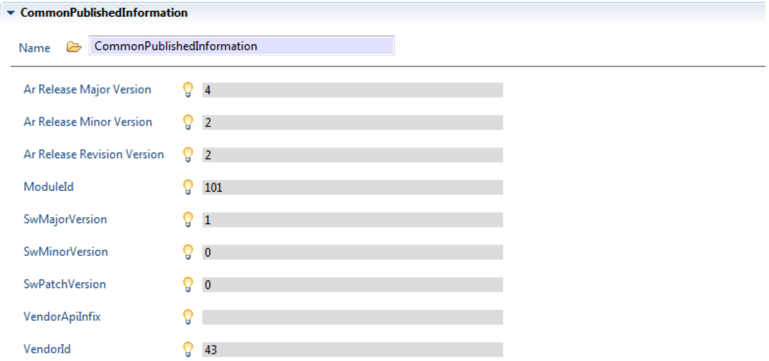
The parameter represents the different type of reset that a Micro supports. This parameter is referenced by the parameter EcuMResetReason in the ECU State manager module.

Table 4-32. Attribute McuResetReason (McuResetReasonConf) detailed description

Property	Value
Type	INTEGER_LABEL
Origin	AUTOSAR_ECUC
Symbolic Name	true
Default	0
Invalid	Range <=255 >=0

4.8 Form CommonPublishedInformation

Common container, aggregated by all modules. It contains published information about vendor and versions.



CommonPublishedInformation

Name	CommonPublishedInformation
Ar Release Major Version	4
Ar Release Minor Version	2
Ar Release Revision Version	2
ModuleId	101
SwMajorVersion	1
SwMinorVersion	0
SwPatchVersion	0
VendorApiInfix	
VendorId	43

Figure 4-8. Tresos Plugin snapshot for CommonPublishedInformation form.

4.8.1 ArReleaseMajorVersion (CommonPublishedInformation)

Major version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-33. Attribute ArReleaseMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range >=4 <=4

4.8.2 ArReleaseMinorVersion (CommonPublishedInformation)

Minor version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-34. Attribute ArReleaseMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.8.3 ArReleaseRevisionVersion (CommonPublishedInformation)

Revision version number of AUTOSAR specification on which the appropriate implementation is based on.

Table 4-35. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description

Property	Value
Label	AUTOSAR Release Revision Version
Type	INTEGER_LABEL

Table continues on the next page...

Table 4-35. Attribute ArReleaseRevisionVersion (CommonPublishedInformation) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	2
Invalid	Range >=2 <=2

4.8.4 ModuleId (CommonPublishedInformation)

Module ID of this module from Module List.

Table 4-36. Attribute ModuleId (CommonPublishedInformation) detailed description

Property	Value
Label	Module Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	101
Invalid	Range >=101 <=101

4.8.5 SwMajorVersion (CommonPublishedInformation)

Major version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-37. Attribute SwMajorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Major Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range >=1 <=1

4.8.6 SwMinorVersion (CommonPublishedInformation)

Minor version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-38. Attribute SwMinorVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Minor Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.8.7 SwPatchVersion (CommonPublishedInformation)

Patch level version number of the vendor specific implementation of the module. The numbering is vendor specific.

Table 4-39. Attribute SwPatchVersion (CommonPublishedInformation) detailed description

Property	Value
Label	Software Patch Version
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range >=0 <=0

4.8.8 VendorApiInfix (CommonPublishedInformation)

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name. This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows: <ModuleName>_>VendorId>_<VendorApiInfix><Api name from SWS>. E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write. This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

Table 4-40. Attribute VendorApiInfix (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Api Infix
Type	STRING_LABEL
Origin	Custom
Symbolic Name	false
Default	
Enable	false

4.8.9 VendorId (CommonPublishedInformation)

Vendor ID of the dedicated implementation of this module according to the AUTOSAR vendor list.

Table 4-41. Attribute VendorId (CommonPublishedInformation) detailed description

Property	Value
Label	Vendor Id
Type	INTEGER_LABEL
Origin	Custom
Symbolic Name	false
Default	43
Invalid	Range >=43 <=43

4.9 Form McuModuleConfiguration

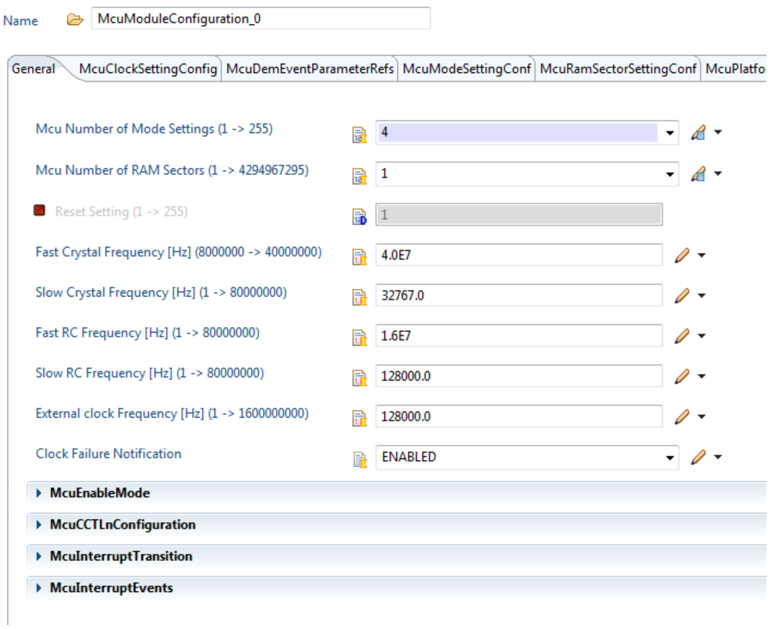
This container contains the configuration for the MCU driver.

Included forms :

- Form McuDemEventParameterRefs

- Form McuEnableMode
- Form McuCoreConfiguration
- Form McuInterruptTransition
- Form McuInterruptEvents
- Form McuPlatformFlashConfig
- Form McuResetConfig
- Form McuPowerControl

- Form McuClockSettingConfig
- Form McuModeSettingConf
- Form McuRamSectorSettingConf
- Form McuRunConfig
- Form McuLowPowerConfig
- Form McuPeripheral



Name	McuModuleConfiguration_0
Mcu Number of Mode Settings (1 -> 255)	4
Mcu Number of RAM Sectors (1 -> 4294967295)	1
Reset Setting (1 -> 255)	1
Fast Crystal Frequency [Hz] (8000000 -> 40000000)	4.0E7
Slow Crystal Frequency [Hz] (1 -> 80000000)	32767.0
Fast RC Frequency [Hz] (1 -> 80000000)	1.6E7
Slow RC Frequency [Hz] (1 -> 80000000)	128000.0
External clock Frequency [Hz] (1 -> 1600000000)	128000.0
Clock Failure Notification	ENABLED

McuEnableMode

McuCCTLnConfiguration

McuInterruptTransition

McuInterruptEvents

Figure 4-9. Tresos Plugin snapshot for McuModuleConfiguration form.

4.9.1 McuNumberOfMcuModes (McuModuleConfiguration)

This parameter shall represent the number of Modes available for the MCU (from "McuModeSettingConf" list). CalculationFormula = Number of configured "McuModeSettingConf". This parameter is not used.

Table 4-42. Attribute McuNumberOfMcuModes (McuModuleConfiguration) detailed description

Property	Value
Label	Mcu Number of Mode Settings
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Invalid	Range <=255 >=1

4.9.2 McuRamSectors (McuModuleConfiguration)

This parameter shall represent the number of RAM sectors available for the MCU (from "McuRamSectorSettingConf" list). CalculationFormula = Number of configured "McuRamSectorSettingConf". This parameter is not used.

Table 4-43. Attribute McuRamSectors (McuModuleConfiguration) detailed description

Property	Value
Label	Mcu Number of RAM Sectors
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Invalid	Range <=4294967295 >=0

4.9.3 McuResetSetting (McuModuleConfiguration)

This parameters applies to the function Mcu_PerformReset(), which performs a microcontroller reset using the hardware feature of the microcontroller. Note: This parameter is not used by the current Implementation. Software Reset occurs when Mcu_PerformReset() function is called. This parameter is not used.

Table 4-44. Attribute McuResetSetting (McuModuleConfiguration) detailed description

Property	Value
Label	Reset Setting
Type	INTEGER

Table continues on the next page...

Table 4-44. Attribute McuResetSetting (McuModuleConfiguration) detailed description (continued)

Property	Value
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1
Invalid	Range <=255 >=1

4.9.4 McuFastCrystalFrequencyHz (McuModuleConfiguration)

Fast Crystal Frequency [Hz].

Note: Implementation Specific Parameter.

Table 4-45. Attribute McuFastCrystalFrequencyHz (McuModuleConfiguration) detailed description

Property	Value
Label	Fast Crystal Frequency [Hz]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	40000000
Invalid	Range <=40000000 >=8000000

4.9.5 McuSlowCrystalFrequencyHz (McuModuleConfiguration)

Slow Crystal Frequency [Hz].

Note: Implementation Specific Parameter.

Table 4-46. Attribute McuSlowCrystalFrequencyHz (McuModuleConfiguration) detailed description

Property	Value
Label	Slow Crystal Frequency [Hz]

Table continues on the next page...

Table 4-46. Attribute McuSlowCrystalFrequencyHz (McuModuleConfiguration) detailed description (continued)

Property	Value
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	32767
Invalid	Range <=80000000 >=1

4.9.6 McuFastIrcFrequencyHz (McuModuleConfiguration)

Fast RC Oscillator Frequency [Hz].

Note: Implementation Specific Parameter.

Table 4-47. Attribute McuFastIrcFrequencyHz (McuModuleConfiguration) detailed description

Property	Value
Label	Fast RC Frequency [Hz]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	16000000
Invalid	Range <=80000000 >=1

4.9.7 McuSlowIrcFrequencyHz (McuModuleConfiguration)

Slow RC Oscillator Frequency [Hz].

Note: Implementation Specific Parameter.

Table 4-48. Attribute McuSlowIrcFrequencyHz (McuModuleConfiguration) detailed description

Property	Value
Label	Slow RC Frequency [Hz]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	128000
Invalid	Range <=80000000 >=1

4.9.8 McuExternalClockFrequencyHz (McuModuleConfiguration)

External clock Frequency [Hz].

Note: Implementation Specific Parameter.

Table 4-49. Attribute McuExternalClockFrequencyHz (McuModuleConfiguration) detailed description

Property	Value
Label	External clock Frequency [Hz]
Type	FLOAT
Origin	Custom
Symbolic Name	false
Default	128000
Invalid	Range <=1600000000 >=1

4.9.9 McuClockSrcFailureNotification (McuModuleConfiguration)

Enables/Disables clock failure notification. In case this feature is not supported by HW the setting should be disabled.

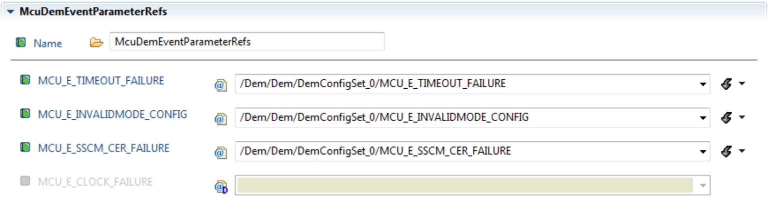
Table 4-50. Attribute McuClockSrcFailureNotification (McuModuleConfiguration) detailed description

Property	Value
Label	Clock Failure Notification
Type	ENUMERATION
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	ENABLED
Range	ENABLED DISABLED

4.9.10 Form McuDemEventParameterRefs

Container for the references to DemEventParameter elements which shall be invoked using the API Dem_ReportErrorStatus API in case the corresponding error occurs. The EventId is taken from the referenced DemEventParameter's DemEventId value. The standardized errors are provided in the container and can be extended by vendor specific error references.

Is included by form : Form McuModuleConfiguration



McuDemEventParameterRefs

Name	McuDemEventParameterRefs
MCU_E_TIMEOUT_FAILURE	/Dem/Dem/DemConfigSet_0/MCU_E_TIMEOUT_FAILURE
MCU_E_INVALIDMODE_CONFIG	/Dem/Dem/DemConfigSet_0/MCU_E_INVALIDMODE_CONFIG
MCU_E_SSCM_CER_FAILURE	/Dem/Dem/DemConfigSet_0/MCU_E_SSCM_CER_FAILURE
MCU_E_CLOCK_FAILURE	

Figure 4-10. Tresos Plugin snapshot for McuDemEventParameterRefs form.

4.9.10.1 MCU_E_TIMEOUT_FAILURE (McuDemEventParameterRefs)

Reference to configured DEM event to report Timeout failure.

Table 4-51. Attribute MCU_E_TIMEOUT_FAILURE (McuDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom

4.9.10.2 MCU_E_INVALIDMODE_CONFIG (McuDemEventParameterRefs)

Reference to configured DEM event to report a MC_ME invalid mode config event.

Table 4-52. Attribute MCU_E_INVALIDMODE_CONFIG (McuDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom

4.9.10.3 MCU_E_SSCM_CER_FAILURE (McuDemEventParameterRefs)

Reference to configured DEM event to report a SSCM Configuration Error.

Table 4-53. Attribute MCU_E_SSCM_CER_FAILURE (McuDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom

4.9.10.4 MCU_E_CLOCK_FAILURE (McuDemEventParameterRefs)

Reference to configured DEM event to report Clock source failure.

Table 4-54. Attribute MCU_E_CLOCK_FAILURE (McuDemEventParameterRefs) detailed description

Property	Value
Type	SYMBOLIC-NAME-REFERENCE
Origin	AUTOSAR_ECUC

4.9.11 Form McuEnableMode

This container allows a way to disable the device modes which are not required for a given device. Configuration for ME_ME register (Module Entry Module). RESET_DEST, RUN0, DRUN, SAFE, RESET modes are always enabled and can not be disabled by the user. User Modes configurable: STOP0, STANDBY0, RUN3, RUN2, RUN1, TEST. Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration

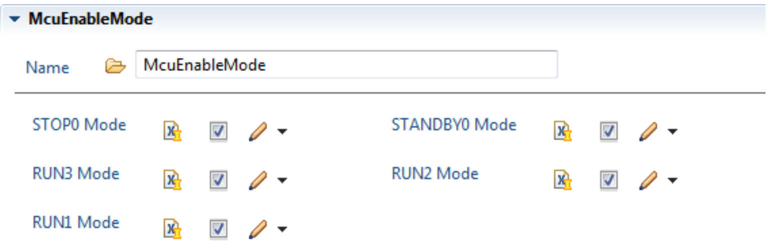


Figure 4-11. Tresos Plugin snapshot for McuEnableMode form.

4.9.11.1 McuModeStop0 (McuEnableMode)

Check if you want enable the STOP0 mode. Refer to ME_ME[STOP0] bit. Note: Implementation Specific Parameter.

Table 4-55. Attribute McuModeStop0 (McuEnableMode) detailed description

Property	Value
Label	STOP0 Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.11.2 McuModeStandBy0 (McuEnableMode)

Check if you want enable the STANDBY0 mode. Refer to ME_ME[STANDBY0] bit. Note: Implementation Specific Parameter.

Table 4-56. Attribute McuModeStandBy0 (McuEnableMode) detailed description

Property	Value
Label	STANDBY0 Mode

Table continues on the next page...

Table 4-56. Attribute McuModeStandBy0 (McuEnableMode) detailed description (continued)

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.11.3 McuModeRun3 (McuEnableMode)

Check if you want enable the RUN3 mode. Refer to ME_ME[RUN3] bit. Note: Implementation Specific Parameter.

Table 4-57. Attribute McuModeRun3 (McuEnableMode) detailed description

Property	Value
Label	RUN3 Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.11.4 McuModeRun2 (McuEnableMode)

Check if you want enable the RUN2 mode. Refer to ME_ME[RUN2] bit. Note: Implementation Specific Parameter.

Table 4-58. Attribute McuModeRun2 (McuEnableMode) detailed description

Property	Value
Label	RUN2 Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.11.5 McuModeRun1 (McuEnableMode)

Check if you want enable the RUN1 mode. Refer to ME_ME[RUN1] bit. Note: Implementation Specific Parameter.

Table 4-59. Attribute McuModeRun1 (McuEnableMode) detailed description

Property	Value
Label	RUN1 Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12 Form McuCoreConfiguration

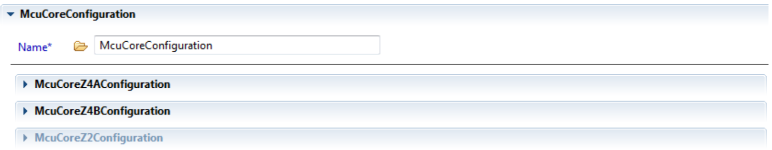
This control allows to enable multi-core.

Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration

Included forms :

- Form McuCoreZ4AConfiguration
- Form McuCoreZ4BConfiguration
- Form McuCoreZ2Configuration



Name* McuCoreConfiguration

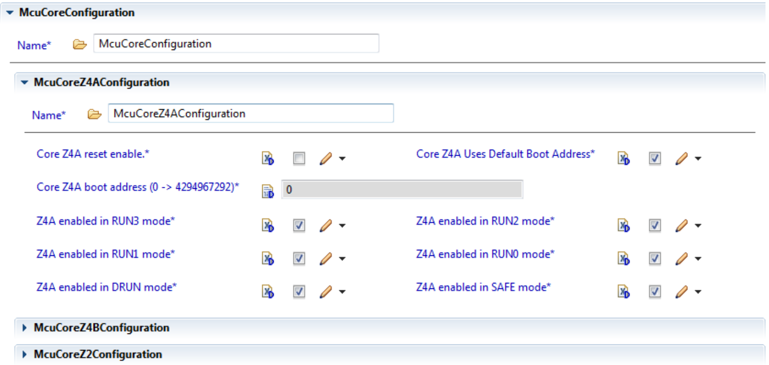
Figure 4-12. Tresos Plugin snapshot for McuCoreConfiguration form.

4.9.12.1 Form McuCoreZ4AConfiguration

This control allows to configure the value of Core Z4A registers.

Note: Implementation Specific Parameter.

Is included by form : Form McuCoreConfiguration



McuCoreConfiguration

Name* McuCoreConfiguration

McuCoreZ4AConfiguration

Name* McuCoreZ4AConfiguration

Core Z4A reset enable.* ☐

Core Z4A Uses Default Boot Address* ☑

Core Z4A boot address (0 -> 4294967292)* 0

Z4A enabled in RUN3 mode* ☑

Z4A enabled in RUN2 mode* ☑

Z4A enabled in RUN1 mode* ☑

Z4A enabled in RUN0 mode* ☑

Z4A enabled in DRUN mode* ☑

Z4A enabled in SAFE mode* ☑

McuCoreZ4BConfiguration

McuCoreZ2Configuration

Figure 4-13. Tresos Plugin snapshot for McuCoreZ4AConfiguration form.

4.9.12.1.1 McuCoreReset (McuCoreZ4AConfiguration)

Set this to TRUE to enable core Z4A Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-60. Attribute McuCoreReset (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Core Z4A reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.1.2 McuUseDefaultBootAddrEnable (McuCoreZ4AConfiguration)

1 - Core Z4A Boot Address is not updated.

0 - Core Z4A Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-61. Attribute McuUseDefaultBootAddrEnable (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Core Z4A Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-61. Attribute McuUseDefaultBootAddrEnable (McuCoreZ4AConfiguration) detailed description (continued)

Property	Value
Default	true

4.9.12.1.3 McuBootAddress (McuCoreZ4AConfiguration)

Boot address for core Z4A after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

Table 4-62. Attribute McuBootAddress (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Core Z4A boot address
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.12.1.4 McuCCTL1Run3 (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in RUN3 mode

Note: Implementation Specific Parameter.

Table 4-63. Attribute McuCCTL1Run3 (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Z4A enabled in RUN3 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.1.5 McuCCTL1Run2 (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in RUN2 mode

Note: Implementation Specific Parameter.

Table 4-64. Attribute McuCCTL1Run2 (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Z4A enabled in RUN2 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.1.6 McuCCTL1Run1 (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in RUN1 mode

Note: Implementation Specific Parameter.

Table 4-65. Attribute McuCCTL1Run1 (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Z4A enabled in RUN1 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.1.7 McuCCTL1Run0 (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in RUN0 mode

Note: Implementation Specific Parameter.

Table 4-66. Attribute McuCCTL1Run0 (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Z4A enabled in RUN0 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.1.8 McuCCTL1Drun (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in DRUN mode

Note: Implementation Specific Parameter.

Table 4-67. Attribute McuCCTL1Drun (McuCoreZ4AConfiguration) detailed description

Property	Value
Label	Z4A enabled in DRUN mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.1.9 McuCCTL1Safe (McuCoreZ4AConfiguration)

Check this bit to enable core Z4A in SAFE mode

Note: Implementation Specific Parameter.

Table 4-68. Attribute McuCCTL1Safe (McuCoreZ4AConfiguration) detailed description

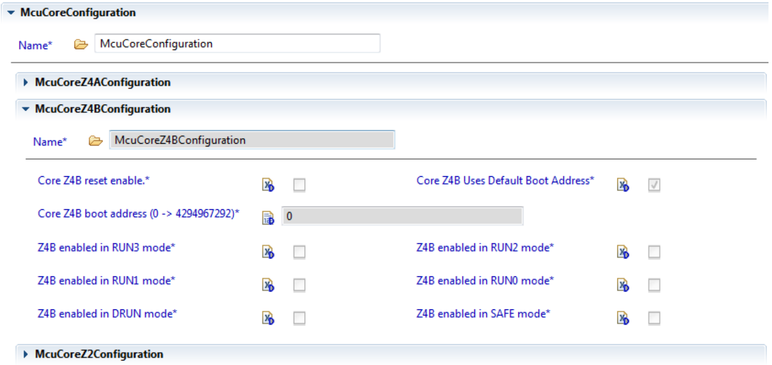
Property	Value
Label	Z4A enabled in SAFE mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.12.2 Form McuCoreZ4BConfiguration

This control allows to configure the value of Core Z4B registers.

Note: Implementation Specific Parameter.

Is included by form : Form McuCoreConfiguration



Name* McuCoreConfiguration

Name* McuCoreZ4BConfiguration

Core Z4B reset enable.*	□	Core Z4B Uses Default Boot Address*	☑
Core Z4B boot address (0 -> 4294967292)*	0		
Z4B enabled in RUN3 mode*	□	Z4B enabled in RUN2 mode*	□
Z4B enabled in RUN1 mode*	□	Z4B enabled in RUN0 mode*	□
Z4B enabled in DRUN mode*	□	Z4B enabled in SAFE mode*	□

Figure 4-14. Tresos Plugin snapshot for McuCoreZ4BConfiguration form.

4.9.12.2.1 McuCoreReset (McuCoreZ4BConfiguration)

Set this to TRUE to enable core Z4B Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-69. Attribute McuCoreReset (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Core Z4B reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.2 McuUseDefaultBootAddrEnable (McuCoreZ4BConfiguration)

1 - Core Z4B Boot Address is not updated.

0 - Core Z4B Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-70. Attribute McuUseDefaultBootAddrEnable (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Core Z4B Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-70. Attribute McuUseDefaultBootAddrEnable (McuCoreZ4BConfiguration) detailed description (continued)

Property	Value
Default	true

4.9.12.2.3 McuBootAddress (McuCoreZ4BConfiguration)

Boot address for core Z4B after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

Table 4-71. Attribute McuBootAddress (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Core Z4B boot address
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.12.2.4 McuCCTL2Run3 (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in RUN3 mode

Note: Implementation Specific Parameter.

Table 4-72. Attribute McuCCTL2Run3 (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Z4B enabled in RUN3 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.5 McuCCTL2Run2 (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in RUN2 mode

Note: Implementation Specific Parameter.

Table 4-73. Attribute McuCCTL2Run2 (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Z4B enabled in RUN2 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.6 McuCCTL2Run1 (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in RUN1 mode

Note: Implementation Specific Parameter.

Table 4-74. Attribute McuCCTL2Run1 (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Z4B enabled in RUN1 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.7 McuCCTL2Run0 (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in RUN0 mode

Note: Implementation Specific Parameter.

Table 4-75. Attribute McuCCTL2Run0 (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Z4B enabled in RUN0 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.8 McuCCTL2Drun (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in DRUN mode

Note: Implementation Specific Parameter.

Table 4-76. Attribute McuCCTL2Drun (McuCoreZ4BConfiguration) detailed description

Property	Value
Label	Z4B enabled in DRUN mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.2.9 McuCCTL2Safe (McuCoreZ4BConfiguration)

Check this bit to enable core Z4B in SAFE mode

Note: Implementation Specific Parameter.

Table 4-77. Attribute McuCCTL2Safe (McuCoreZ4BConfiguration) detailed description

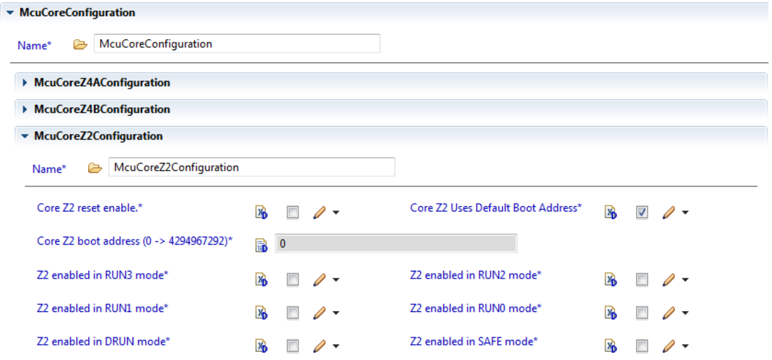
Property	Value
Label	Z4B enabled in SAFE mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3 Form McuCoreZ2Configuration

This control allows to configure the value of Core Z2 registers.

Note: Implementation Specific Parameter.

Is included by form : Form McuCoreConfiguration



▾ McuCoreConfiguration

Name* McuCoreConfiguration

▸ McuCoreZ4AConfiguration

▸ McuCoreZ4BConfiguration

▾ McuCoreZ2Configuration

Name* McuCoreZ2Configuration

Core Z2 reset enable.* □

Core Z2 Uses Default Boot Address* ☑

Core Z2 boot address (0 -> 4294967292)* 0

Z2 enabled in RUN3 mode* □

Z2 enabled in RUN2 mode* □

Z2 enabled in RUN1 mode* □

Z2 enabled in RUN0 mode* □

Z2 enabled in DRUN mode* □

Z2 enabled in SAFE mode* □

Figure 4-15. Tresos Plugin snapshot for McuCoreZ2Configuration form.

4.9.12.3.1 McuCoreReset (McuCoreZ2Configuration)

Set this to TRUE to enable core Z2 Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-78. Attribute McuCoreReset (McuCoreZ2Configuration) detailed description

Property	Value
Label	Core Z2 reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.2 McuUseDefaultBootAddrEnable (McuCoreZ2Configuration)

1 - Core Z2 Boot Address is not updated.

0 - Core Z2 Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-79. Attribute McuUseDefaultBootAddrEnable (McuCoreZ2Configuration) detailed description

Property	Value
Label	Core Z2 Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

Table continues on the next page...

Table 4-79. Attribute McuUseDefaultBootAddrEnable (McuCoreZ2Configuration) detailed description (continued)

Property	Value
Default	true

4.9.12.3.3 McuBootAddress (McuCoreZ2Configuration)

Boot address for core Z2 after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

Table 4-80. Attribute McuBootAddress (McuCoreZ2Configuration) detailed description

Property	Value
Label	Core Z2 boot address
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.12.3.4 McuCCTL3Run3 (McuCoreZ2Configuration)

Check this bit to enable core Z2 in RUN3 mode

Note: Implementation Specific Parameter.

Table 4-81. Attribute McuCCTL3Run3 (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in RUN3 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.5 McuCCTL3Run2 (McuCoreZ2Configuration)

Check this bit to enable core Z2 in RUN2 mode

Note: Implementation Specific Parameter.

Table 4-82. Attribute McuCCTL3Run2 (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in RUN2 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.6 McuCCTL3Run1 (McuCoreZ2Configuration)

Check this bit to enable core Z2 in RUN1 mode

Note: Implementation Specific Parameter.

Table 4-83. Attribute McuCCTL3Run1 (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in RUN1 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.7 McuCCTL3Run0 (McuCoreZ2Configuration)

Check this bit to enable core Z2 in RUN0 mode

Note: Implementation Specific Parameter.

Table 4-84. Attribute McuCCTL3Run0 (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in RUN0 mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.8 McuCCTL3Drun (McuCoreZ2Configuration)

Check this bit to enable core Z2 in DRUN mode

Note: Implementation Specific Parameter.

Table 4-85. Attribute McuCCTL3Drun (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in DRUN mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.12.3.9 McuCCTL3Safe (McuCoreZ2Configuration)

Check this bit to enable core Z2 in SAFE mode

Note: Implementation Specific Parameter.

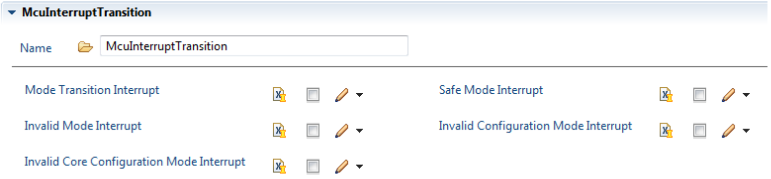
Table 4-86. Attribute McuCCTL3Safe (McuCoreZ2Configuration) detailed description

Property	Value
Label	Z2 enabled in SAFE mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.13 Form McuInterruptTransition

Configuration of ME_IM register. Interrupt Mask Register (ME_IM): This register controls whether an event generates an interrupt or not. Note: Implementation specific Container.

Is included by form : Form McuModuleConfiguration



McuInterruptTransition

Name McuInterruptTransition

Mode Transition Interrupt □

Safe Mode Interrupt □

Invalid Mode Interrupt □

Invalid Configuration Mode Interrupt □

Invalid Core Configuration Mode Interrupt □

Figure 4-16. Tresos Plugin snapshot for McuInterruptTransition form.

4.9.13.1 McuTransitionComplete (McuInterruptTransition)

Mode transition complete interrupt mask.

Configure the ME_IM[M_MTC] field register.

- 0 Mode transition complete interrupt is masked.

- 1 Mode transition complete interrupt is enabled.

Whenever the system fully completes a mode transition (i.e. the S_MTRANS bit of ME_GS register transits from '1' to '0'), the interrupt pending bit I_MTC of the ME_IS register is set, and an interrupt request is generated if the mask bit M_MTC of the ME_IM register is '1'. The interrupt bit I_MTC is not set when entering low-power modes STANDBY0, LPU_RUN and STOP0 in order to avoid the same event requesting the immediate exit of these low-power modes.

Note: Implementation Specific Parameter.

Table 4-87. Attribute McuTransitionComplete (McuInterruptTransition) detailed description

Property	Value
Label	Mode Transition Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.13.2 McuSafeMode (McuInterruptTransition)

SAFE mode interrupt mask.

Configure the ME_IM[M_SAFE] field register.

- 0 SAFE mode interrupt is masked.

- 1 SAFE mode interrupt is enabled.

Whenever the system enters the SAFE mode as a result of a SAFE mode request from the MC_RGM due to a hardware failure, the interrupt pending bit I_SAFE of the ME_IS register is set, and an interrupt is generated if the mask bit M_SAFE of ME_IM register is '1'.

Note: Implementation Specific Parameter.

Table 4-88. Attribute McuSafeMode (McuInterruptTransition) detailed description

Property	Value
Label	Safe Mode Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.13.3 McuInvalidMode (McuInterruptTransition)

Invalid mode interrupt mask.

Configure the ME_IM[M_IMODE] field register.

- 0 Invalid mode interrupt is masked.

- 1 Invalid mode interrupt is enabled.

Whenever an invalid mode request is detected, the interrupt pending bit I_IMODE of the ME_IS register is set, and an interrupt request is generated if the mask bit M_IMODE of the ME_IM register is '1'.

Note: Implementation Specific Parameter.

Table 4-89. Attribute McuInvalidMode (McuInterruptTransition) detailed description

Property	Value
Label	Invalid Mode Interrupt

Table continues on the next page...

Table 4-89. Attribute McuInvalidMode (McuInterruptTransition) detailed description (continued)

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.13.4 McuInvalidConfiguration (McuInterruptTransition)

Invalid mode configuration interrupt mask.

Configure the ME_IM[M_ICONF] field register.

- 0 Invalid mode interrupt is masked.

- 1 Invalid mode interrupt is enabled.

Whenever a write operation is attempted to the ME_<mode>_MC registers violating the protection rules, the interrupt pending bit I_ICONF of the ME_IS register is set and an interrupt request is generated if the mask bit M_ICONF of ME_IM register is '1'.

Protection of mode configuration registers:

- If the 16 MHz int. RC osc. is selected as the system clock, IRCOSC must be on.

- If the 4-40 MHz crystal osc. clock is selected as the system clock, XOSC must be on.

- If the system PLL clock is selected as the system clock, PLL0 must be on.

- If PLL0 is on, XOSC must also be on.

- If PLL1 is on, XOSC must also be on.

- Configuration "00" for the FLAON bit field is reserved.

- System clock configurations marked as "reserved" may not be selected.

- Configuration "1111" for the SYSCLK bit field is allowed only for the TEST mode, and only in this case may all system clock sources be turned off.

Note: Implementation Specific Parameter.

Table 4-90. Attribute McuInvalidConfiguration (McuInterruptTransition) detailed description

Property	Value
Label	Invalid Configuration Mode Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.13.5 McuInvalidCoreConfiguration (McuInterruptTransition)

Invalid mode configuration (core configuration) interrupt mask.

Configure the ME_IM[M_ICONF_CC] field register.

- 0 Invalid mode interrupt is masked.

- 1 Invalid mode interrupt is enabled.

This event is generated if a write access to one of the ME_CCTLn

registers is attempted while a mode transition is in progress.

Note: Implementation Specific Parameter.

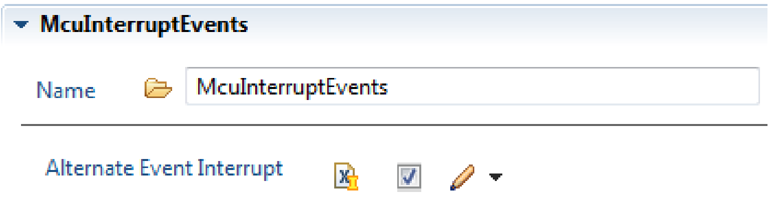
Table 4-91. Attribute McuInvalidCoreConfiguration (McuInterruptTransition) detailed description

Property	Value
Label	Invalid Core Configuration Mode Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.14 Form McuInterruptEvents

Configuration for different interrupts handled by MCU. Note: Implementation specific Container.

Is included by form : Form McuModuleConfiguration



McuInterruptEvents

Name: McuInterruptEvents

Figure 4-17. Tresos Plugin snapshot for McuInterruptEvents form.

4.9.14.1 McuAlternateResetEvent (McuInterruptEvents)

Some events can generate an interrupt from MC_RGM.

Note: Implementation Specific Parameter.

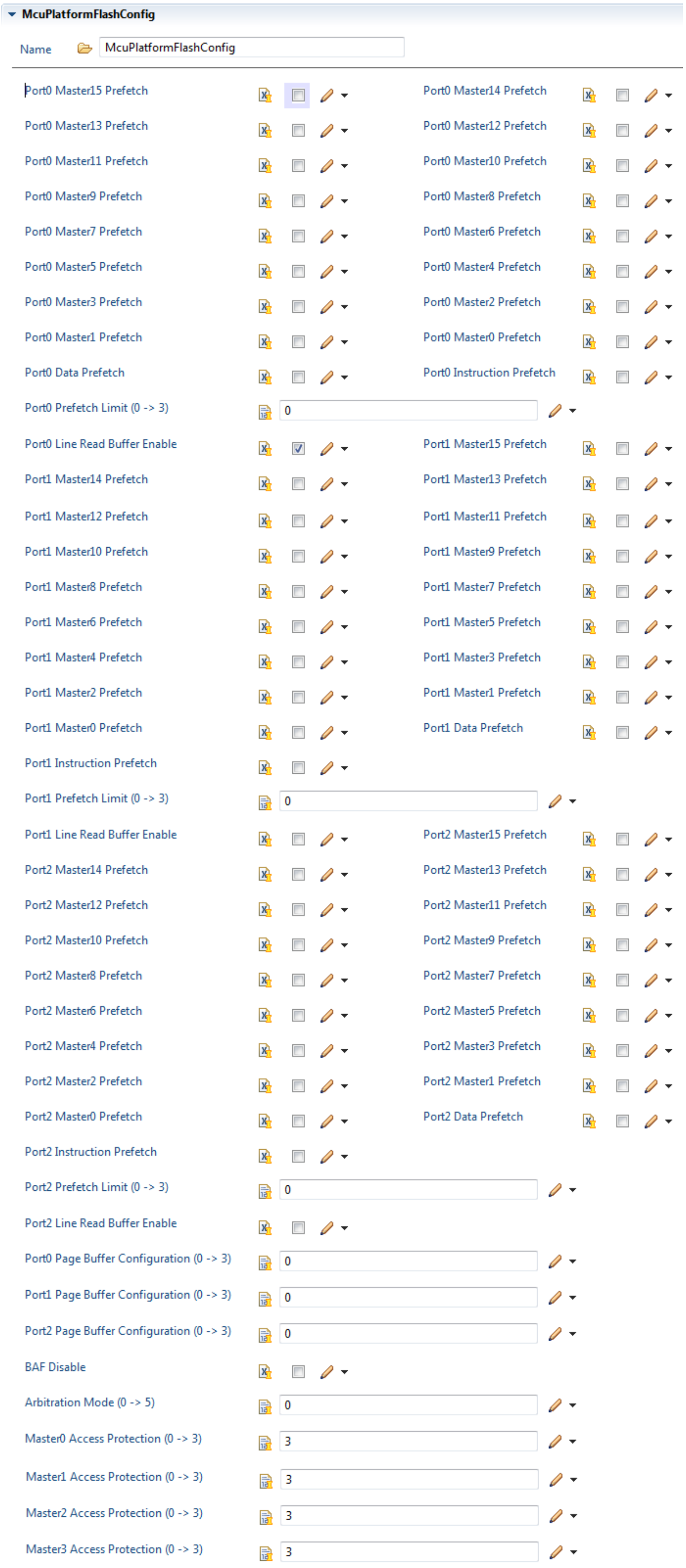
Table 4-92. Attribute McuAlternateResetEvent (McuInterruptEvents) detailed description

Property	Value
Label	Alternate Event Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15 Form McuPlatformFlashConfig

This container contains the specific configuration (parameters) of the flash memory array. Note: Implementation Specific Container.

Is included by form : Form McuModuleConfiguration



McuPlatformFlashConfig

Name: McuPlatformFlashConfig

Port0 Master15 Prefetch	☐	Port0 Master14 Prefetch	☐
Port0 Master13 Prefetch	☐	Port0 Master12 Prefetch	☐
Port0 Master11 Prefetch	☐	Port0 Master10 Prefetch	☐
Port0 Master9 Prefetch	☐	Port0 Master8 Prefetch	☐
Port0 Master7 Prefetch	☐	Port0 Master6 Prefetch	☐
Port0 Master5 Prefetch	☐	Port0 Master4 Prefetch	☐
Port0 Master3 Prefetch	☐	Port0 Master2 Prefetch	☐
Port0 Master1 Prefetch	☐	Port0 Master0 Prefetch	☐
Port0 Data Prefetch	☐	Port0 Instruction Prefetch	☐
Port0 Prefetch Limit (0 -> 3)	0		
Port0 Line Read Buffer Enable	☑	Port1 Master15 Prefetch	☐
Port1 Master14 Prefetch	☐	Port1 Master13 Prefetch	☐
Port1 Master12 Prefetch	☐	Port1 Master11 Prefetch	☐
Port1 Master10 Prefetch	☐	Port1 Master9 Prefetch	☐
Port1 Master8 Prefetch	☐	Port1 Master7 Prefetch	☐
Port1 Master6 Prefetch	☐	Port1 Master5 Prefetch	☐
Port1 Master4 Prefetch	☐	Port1 Master3 Prefetch	☐
Port1 Master2 Prefetch	☐	Port1 Master1 Prefetch	☐
Port1 Master0 Prefetch	☐	Port1 Data Prefetch	☐
Port1 Instruction Prefetch	☐		
Port1 Prefetch Limit (0 -> 3)	0		
Port1 Line Read Buffer Enable	☐	Port2 Master15 Prefetch	☐
Port2 Master14 Prefetch	☐	Port2 Master13 Prefetch	☐
Port2 Master12 Prefetch	☐	Port2 Master11 Prefetch	☐
Port2 Master10 Prefetch	☐	Port2 Master9 Prefetch	☐
Port2 Master8 Prefetch	☐	Port2 Master7 Prefetch	☐
Port2 Master6 Prefetch	☐	Port2 Master5 Prefetch	☐
Port2 Master4 Prefetch	☐	Port2 Master3 Prefetch	☐
Port2 Master2 Prefetch	☐	Port2 Master1 Prefetch	☐
Port2 Master0 Prefetch	☐	Port2 Data Prefetch	☐
Port2 Instruction Prefetch	☐		
Port2 Prefetch Limit (0 -> 3)	0		
Port2 Line Read Buffer Enable	☐		
Port0 Page Buffer Configuration (0 -> 3)	0		
Port1 Page Buffer Configuration (0 -> 3)	0		
Port2 Page Buffer Configuration (0 -> 3)	0		
BAF Disable	☐		
Arbitration Mode (0 -> 5)	0		
Master0 Access Protection (0 -> 3)	3		
Master1 Access Protection (0 -> 3)	3		
Master2 Access Protection (0 -> 3)	3		
Master3 Access Protection (0 -> 3)	3		

Figure 4-18. Tresos Plugin snapshot for McuPlatformFlashConfig form.

4.9.15.1 McuPort0Master15Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-93. Attribute McuPort0Master15Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master15 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.2 McuPort0Master14Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-94. Attribute McuPort0Master14Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master14 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.3 McuPort0Master13Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-95. Attribute McuPort0Master13Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master13 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.4 McuPort0Master12Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-96. Attribute McuPort0Master12Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master12 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.5 McuPort0Master11Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-97. Attribute McuPort0Master11Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master11 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.6 McuPort0Master10Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-98. Attribute McuPort0Master10Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master10 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.7 McuPort0Master9Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-99. Attribute McuPort0Master9Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master9 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.8 McuPort0Master8Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-100. Attribute McuPort0Master8Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master8 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.9 McuPort0Master7Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-101. Attribute McuPort0Master7Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master7 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.10 McuPort0Master6Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-102. Attribute McuPort0Master6Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master6 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.11 McuPort0Master5Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-103. Attribute McuPort0Master5Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master5 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.12 McuPort0Master4Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-104. Attribute McuPort0Master4Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master4 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.13 McuPort0Master3Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-105. Attribute McuPort0Master3Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master3 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.14 McuPort0Master2Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-106. Attribute McuPort0Master2Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master2 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.15 McuPort0Master1Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-107. Attribute McuPort0Master1Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master1 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.16 McuPort0Master0Prefetch (McuPlatformFlashConfig)

PFCR1[P0_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-108. Attribute McuPort0Master0Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Master0 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.17 McuPort0DataPrefetch (McuPlatformFlashConfig)

PFCR1[P0_DPFEN] field register configuration. This field enables or disables prefetching initiated by a data read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by a data read access. 1 - Prefetching may be triggered by any data read access. Note: Implementation Specific Parameter.

Table 4-109. Attribute McuPort0DataPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Data Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.18 McuPort0InstructionPrefetch (McuPlatformFlashConfig)

PFCR1[P0_IPFEN] field register configuration. This bit enables or disables prefetching initiated by an instruction read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by an instruction read access. 1 - Prefetching may be triggered by any instruction read access. Note: Implementation Specific Parameter.

Table 4-110. Attribute McuPort0InstructionPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Instruction Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.19 McuPort0PrefetchLimit (McuPlatformFlashConfig)

PFCR1[P0_PFLIM] field register configuration. This field controls the prefetch algorithm used by the prefetch controller.

This field defines a limit on the maximum number of sequential prefetches which will be attempted between buffer misses.

In all situations when enabled, only a single prefetch is initiated on each buffer miss or hit.

00

-No prefetching is performed.

01

-The referenced line is prefetched on a buffer miss, that is, prefetch on miss.

1x

-The referenced line is prefetched on a buffer miss, or the next sequential page is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit.

Note: Implementation Specific Parameter.

Table 4-111. Attribute McuPort0PrefetchLimit (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Prefetch Limit
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.20 McuPort0ReadBufferEnable (McuPlatformFlashConfig)

PFCR1[P0_BFEN] field register configuration. This bit enables or disables line read buffer hits. It is also used to invalidate the buffers. 0 - The line read buffers are disabled from satisfying read requests, and all buffer valid bits are cleared. 1 - The line read buffers are enabled to satisfy read requests on hits. Buffer valid bits may be set when the buffers are successfully filled. Note: Implementation Specific Parameter.

Table 4-112. Attribute McuPort0ReadBufferEnable (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Line Read Buffer Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.15.21 McuPort1Master15Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-113. Attribute McuPort1Master15Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master15 Prefetch
Type	BOOLEAN

Table continues on the next page...

Table 4-113. Attribute McuPort1Master15Prefetch (McuPlatformFlashConfig) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.22 McuPort1Master14Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-114. Attribute McuPort1Master14Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master14 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.23 McuPort1Master13Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-115. Attribute McuPort1Master13Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master13 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.24 McuPort1Master12Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-116. Attribute McuPort1Master12Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master12 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.25 McuPort1Master11Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-117. Attribute McuPort1Master11Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master11 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.26 McuPort1Master10Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-118. Attribute McuPort1Master10Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master10 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.27 McuPort1Master9Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-119. Attribute McuPort1Master9Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master9 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.28 McuPort1Master8Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-120. Attribute McuPort1Master8Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master8 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.29 McuPort1Master7Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-121. Attribute McuPort1Master7Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master7 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.30 McuPort1Master6Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-122. Attribute McuPort1Master6Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master6 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.31 McuPort1Master5Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-123. Attribute McuPort1Master5Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master5 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.32 McuPort1Master4Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-124. Attribute McuPort1Master4Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master4 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.33 McuPort1Master3Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-125. Attribute McuPort1Master3Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master3 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.34 McuPort1Master2Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-126. Attribute McuPort1Master2Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master2 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.35 McuPort1Master1Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-127. Attribute McuPort1Master1Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master1 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.36 McuPort1Master0Prefetch (McuPlatformFlashConfig)

PFCR2[P1_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-128. Attribute McuPort1Master0Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Master0 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.37 McuPort1DataPrefetch (McuPlatformFlashConfig)

PFCR2[P1_DPFEN] field register configuration. This field enables or disables prefetching initiated by a data read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by a data read access. 1 - Prefetching may be triggered by any data read access. Note: Implementation Specific Parameter.

Table 4-129. Attribute McuPort1DataPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Data Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.38 McuPort1InstructionPrefetch (McuPlatformFlashConfig)

PFCR2[P1_IPFEN] field register configuration. This bit enables or disables prefetching initiated by an instruction read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by an instruction read access. 1 - Prefetching may be triggered by any instruction read access. Note: Implementation Specific Parameter.

Table 4-130. Attribute McuPort1InstructionPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Instruction Prefetch
Type	BOOLEAN

Table continues on the next page...

Table 4-130. Attribute McuPort1InstructionPrefetch (McuPlatformFlashConfig) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.39 McuPort1PrefetchLimit (McuPlatformFlashConfig)

PFCR2[P1_PFLIM] field register configuration. This field controls the prefetch algorithm used by the prefetch controller.

This field defines a limit on the maximum number of sequential prefetches which will be attempted between buffer misses.

In all situations when enabled, only a single prefetch is initiated on each buffer miss or hit.

00

-No prefetching is performed.

01

-The referenced line is prefetched on a buffer miss, that is, prefetch on miss.

1x

-The referenced line is prefetched on a buffer miss, or the next sequential page is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit.

Note: Implementation Specific Parameter.

Table 4-131. Attribute McuPort1PrefetchLimit (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Prefetch Limit
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.40 McuPort1ReadBufferEnable (McuPlatformFlashConfig)

PFCR2[P1_BFEN] field register configuration. This bit enables or disables line read buffer hits. It is also used to invalidate the buffers. 0 - The line read buffers are disabled from satisfying read requests, and all buffer valid bits are cleared. 1 - The line read buffers are enabled to satisfy read requests on hits. Buffer valid bits may be set when the buffers are successfully filled. Note: Implementation Specific Parameter.

Table 4-132. Attribute McuPort1ReadBufferEnable (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Line Read Buffer Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.15.41 McuPort2Master15Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-133. Attribute McuPort2Master15Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master15 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.42 McuPort2Master14Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-134. Attribute McuPort2Master14Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master14 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.43 McuPort2Master13Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-135. Attribute McuPort2Master13Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master13 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.44 McuPort2Master12Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-136. Attribute McuPort2Master12Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master12 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.45 McuPort2Master11Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-137. Attribute McuPort2Master11Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master11 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.46 McuPort2Master10Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-138. Attribute McuPort2Master10Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master10 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.47 McuPort2Master9Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-139. Attribute McuPort2Master9Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master9 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.48 McuPort2Master8Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-140. Attribute McuPort2Master8Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master8 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.49 McuPort2Master7Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-141. Attribute McuPort2Master7Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master7 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.50 McuPort2Master6Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-142. Attribute McuPort2Master6Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master6 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.51 McuPort2Master5Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-143. Attribute McuPort2Master5Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master5 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.52 McuPort2Master4Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-144. Attribute McuPort2Master4Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master4 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.53 McuPort2Master3Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-145. Attribute McuPort2Master3Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master3 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.54 McuPort2Master2Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-146. Attribute McuPort2Master2Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master2 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.55 McuPort2Master1Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-147. Attribute McuPort2Master1Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master1 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.56 McuPort2Master0Prefetch (McuPlatformFlashConfig)

PFCR4[P2_MxPFE] field register configuration. Master x Prefetch Enable. 0 - No prefetching may be triggered by this master. 1 - Prefetching may be triggered by this master. Note: Implementation Specific Parameter.

Table 4-148. Attribute McuPort2Master0Prefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Master0 Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.57 McuPort2DataPrefetch (McuPlatformFlashConfig)

PFCR4[P2_DPFEN] field register configuration. This field enables or disables prefetching initiated by a data read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by a data read access. 1 - Prefetching may be triggered by any data read access. Note: Implementation Specific Parameter.

Table 4-149. Attribute McuPort2DataPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Data Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.58 McuPort2InstructionPrefetch (McuPlatformFlashConfig)

PFCR4[P2_IPFEN] field register configuration. This bit enables or disables prefetching initiated by an instruction read access. This field is cleared by hardware reset. 0 - No prefetching is triggered by an instruction read access. 1 - Prefetching may be triggered by any instruction read access. Note: Implementation Specific Parameter.

Table 4-150. Attribute McuPort2InstructionPrefetch (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Instruction Prefetch
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.59 McuPort2PrefetchLimit (McuPlatformFlashConfig)

PFCR4[P2_PFLIM] field register configuration. This field controls the prefetch algorithm used by the prefetch controller.

This field defines a limit on the maximum number of sequential prefetches which will be attempted between buffer misses.

In all situations when enabled, only a single prefetch is initiated on each buffer miss or hit.

00

-No prefetching is performed.

01

-The referenced line is prefetched on a buffer miss, that is, prefetch on miss.

1x

-The referenced line is prefetched on a buffer miss, or the next sequential page is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit.

Note: Implementation Specific Parameter.

Table 4-151. Attribute McuPort2PrefetchLimit (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Prefetch Limit
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.60 McuPort2ReadBufferEnable (McuPlatformFlashConfig)

PFCR4[P2_BFEN] field register configuration. This bit enables or disables line read buffer hits. It is also used to invalidate the buffers. 0 - The line read buffers are disabled from satisfying read requests, and all buffer valid bits are cleared. 1 - The line read buffers are enabled to satisfy read requests on hits. Buffer valid bits may be set when the buffers are successfully filled. Note: Implementation Specific Parameter.

Table 4-152. Attribute McuPort2ReadBufferEnable (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Line Read Buffer Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.15.61 McuPort0PageBufferConfiguration (McuPlatformFlashConfig)

PFCR3[P0_WCFG] field register configuration. This field controls the configuration of the four page buffers in the PFLASH controller. The buffers can be organized as a "pool" of available resources, or with a fixed partition between instruction and data buffers. 00 - All four buffers are available for any flash access, that is, there is no partitioning of the buffers based on the access type. 01 - Reserved. 10 - The buffers are partitioned into two groups with buffers 0 and 1 allocated for instruction fetches and buffers 2 and 3 for data accesses. 11 - The buffers are partitioned into two groups with buffers 0, 1, 2 allocated for instruction fetches and buffer 3 for data accesses. Note: Implementation Specific Parameter.

Table 4-153. Attribute McuPort0PageBufferConfiguration (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port0 Page Buffer Configuration
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.62 McuPort1PageBufferConfiguration (McuPlatformFlashConfig)

PFCR3[P1_WCFG] field register configuration. This field controls the configuration of the four page buffers in the PFLASH controller. The buffers can be organized as a "pool" of available resources, or with a fixed partition between instruction and data buffers. 00 - All four buffers are available for any flash access, that is, there is no partitioning of the buffers based on the access type. 01 - Reserved. 10 - The buffers are partitioned into two groups with buffers 0 and 1 allocated for instruction fetches and buffers 2 and 3 for data accesses. 11 - The buffers are partitioned into two groups with buffers 0, 1, 2 allocated for instruction fetches and buffer 3 for data accesses. Note: Implementation Specific Parameter.

Table 4-154. Attribute McuPort1PageBufferConfiguration (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port1 Page Buffer Configuration
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.63 McuPort2PageBufferConfiguration (McuPlatformFlashConfig)

PFCR3[P2_WCFG] field register configuration. This field controls the configuration of the four page buffers in the PFLASH controller. The buffers can be organized as a "pool" of available resources, or with a fixed partition between instruction and data buffers. 00 - Both buffers are available for any flash access, that is, there is no partitioning of the buffers based onthe access type. 01 - Reserved. 10 - The buffers are partitioned into two groups with buffer 0 allocated for instruction fetches and buffer 1 for data accesses. 11 - Reserved. Note: Implementation Specific Parameter.

Table 4-155. Attribute McuPort2PageBufferConfiguration (McuPlatformFlashConfig) detailed description

Property	Value
Label	Port2 Page Buffer Configuration
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.64 McuBAFDisable (McuPlatformFlashConfig)

PFCR2[BAF_DIS] field register configuration. BAF Disable. This field controls executable access to the BAF (Boot Assist Flash) region of the flash. Once this field is set, attempted instruction accesses targeting the BAF region are aborted and terminated with a system bus error. 0 - Executable access to the BAF flash region is allowed. 1 - Executable access to the BAF flash region is prohibited. Note: Implementation Specific Parameter.

Table 4-156. Attribute McuBAFDisable (McuPlatformFlashConfig) detailed description

Property	Value
Label	BAF Disable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.15.65 McuFlashArbitrationMode (McuPlatformFlashConfig)

PFLASH_PFCR3[ARBM] field register configuration.

Arbitration Mode. This 4-bit field controls the arbitration of concurrent flash access requests from the three AHB ports of the flash memory controller. In both fixed priority or round-robin modes, write requests are prioritized higher than read requests, and read requests are prioritized higher than speculative prefetch requests whenever multiple ports issue concurrent requests.

Table 4-157. Detailed description.

Value	Meaning
0000	Fixed priority arbitration with AHB p0>p1>p2
0001	Fixed priority arbitration with AHB p0>p2>p1
0010	Fixed priority arbitration with AHB p2>p0>p1
0011	Fixed priority arbitration with AHB p2>p1>p0
0100	Fixed priority arbitration with AHB p1>p0>p2
0101	Fixed priority arbitration with AHB p1>p2>p0

Note: Implementation Specific Parameter.

Table 4-158. Attribute McuFlashArbitrationMode (McuPlatformFlashConfig) detailed description

Property	Value
Label	Arbitration Mode
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.15.66 McuMaster0AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-159. Attribute McuMaster0AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master0 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.67 McuMaster1AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-160. Attribute McuMaster1AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master1 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.68 McuMaster2AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-161. Attribute McuMaster2AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master2 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.69 McuMaster3AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-162. Attribute McuMaster3AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master3 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.70 McuMaster4AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-163. Attribute McuMaster4AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master4 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.71 McuMaster5AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-164. Attribute McuMaster5AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master5 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.72 McuMaster6AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-165. Attribute McuMaster6AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master6 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.73 McuMaster7AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-166. Attribute McuMaster7AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master7 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.74 McuMaster8AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-167. Attribute McuMaster8AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master8 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.75 McuMaster9AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-168. Attribute McuMaster9AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master9 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.76 McuMaster10AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-169. Attribute McuMaster10AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master10 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.77 McuMaster11AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-170. Attribute McuMaster11AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master11 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.78 McuMaster12AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-171. Attribute McuMaster12AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master12 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.79 McuMaster13AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-172. Attribute McuMaster13AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master13 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.80 McuMaster14AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-173. Attribute McuMaster14AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master14 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

4.9.15.81 McuMaster15AccessProt (McuPlatformFlashConfig)

PFAPR[MxAP] field register configuration. Master x Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master number of the initiating module. 00 - No accesses may be performed by this master. 01 - Only read accesses may be performed by this master. 10 - Only write accesses may be performed by this master. 11 - Both read and write accesses may be performed by this master. Note: Implementation Specific Parameter.

Table 4-174. Attribute McuMaster15AccessProt (McuPlatformFlashConfig) detailed description

Property	Value
Label	Master15 Access Protection
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	3

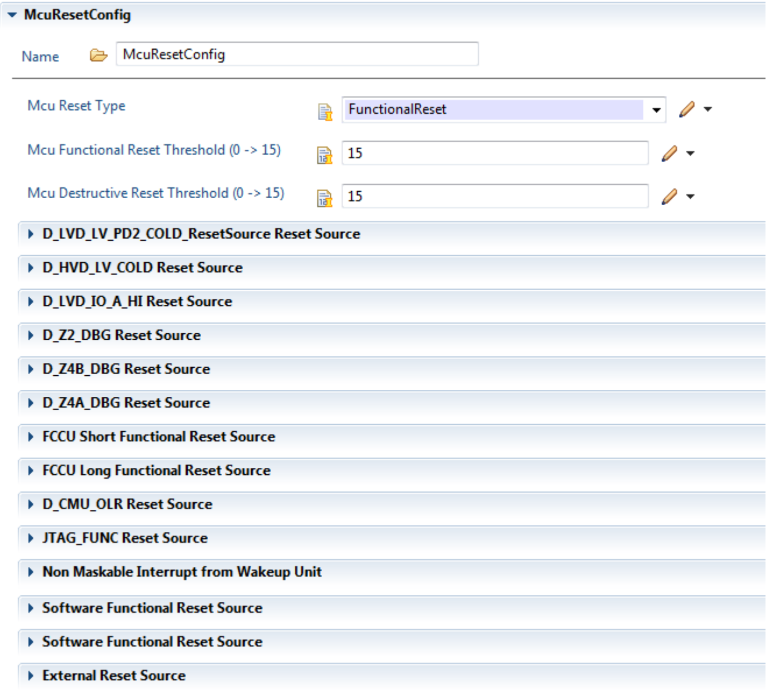
4.9.16 Form McuResetConfig

The reset generation module (MC_RGM) centralizes the different reset sources and manages the reset sequence of the device. Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration

Included forms :

- Form Mcu_D_LVD_LV_PD2_COLD_ResetSource
- Form Mcu_D_HVD_LV_COLD_ResetSource
- Form Mcu_D_LVD_IO_A_HI_ResetSource
- Form Mcu_D_Z2_DBG_ResetSource
- Form Mcu_D_Z4B_DBG_ResetSource
- Form Mcu_D_Z4A_DBG_ResetSource
- Form Mcu_BE_FCCU_SHORT_ResetSource
- Form Mcu_BE_FCCU_LONG_ResetSource
- Form Mcu_D_CMU_OLR_ResetSource
- Form Mcu_D_JTAG_FUNC_ResetSource
- Form Mcu_D_NMI_WKPU_ResetSource
- Form Mcu_SS_SOFT_FUNC_ResetSource
- Form Mcu_BE_HSM_FUNC_ResetSource
- Form Mcu_SS_EXR_ResetSource



McuResetConfig

Name: McuResetConfig

Mcu Reset Type: FunctionalReset

Mcu Functional Reset Threshold (0 -> 15): 15

Mcu Destructive Reset Threshold (0 -> 15): 15

- D_LVD_LV_PD2_COLD_ResetSource Reset Source
- D_HVD_LV_COLD Reset Source
- D_LVD_IO_A_HI Reset Source
- D_Z2_DBG Reset Source
- D_Z4B_DBG Reset Source
- D_Z4A_DBG Reset Source
- FCCU Short Functional Reset Source
- FCCU Long Functional Reset Source
- D_CMU_OLR Reset Source
- JTAG_FUNC Reset Source
- Non Maskable Interrupt from Wakeup Unit
- Software Functional Reset Source
- Software Functional Reset Source
- External Reset Source

Figure 4-19. Tresos Plugin snapshot for McuResetConfig form.

4.9.16.1 McuResetType (McuResetConfig)

This parameter selects the type of the reset to be performed through the McuPerformReset API. A "destructive" reset source is associated with an event related to a critical - usually hardware - error or dysfunction. When a "destructive" reset event occurs, the full reset sequence is applied to the device starting from PHASE0. This resets the full device ensuring a safe start-up state for both digital and analog modules. A "functional" reset source is associated with an event related to a less-critical - usually non-hardware - error or dysfunction. When a "functional" reset event occurs, a partial reset sequence is applied to the device starting from PHASE1. In this case, most digital modules are reset normally, while analog modules or specific digital modules (e.g. debug modules, flash modules) state is preserved.

Table 4-175. Attribute McuResetType (McuResetConfig) detailed description

Property	Value
Label	Mcu Reset Type
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	FunctionalReset
Range	FunctionalReset DestructiveReset

4.9.16.2 McuFunctResetEscThreshold (McuResetConfig)

RGM_FRET[FRET] field configuration. Threshold for a functional reset escalation to a destructive reset. Note: Implementation Specific Parameter.

Table 4-176. Attribute McuFunctResetEscThreshold (McuResetConfig) detailed description

Property	Value
Label	Mcu Functional Reset Threshold
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	15

4.9.16.3 McuDestResetEscThreshold (McuResetConfig)

RGM_FRET[DRET] field configuration.

This register sets the threshold for 'destructive' reset escalation to keeping the chip in the reset state until the next power-on reset triggers a new reset sequence.

Note: Implementation Specific Parameter.

Table 4-177. Attribute McuDestResetEscThreshold (McuResetConfig) detailed description

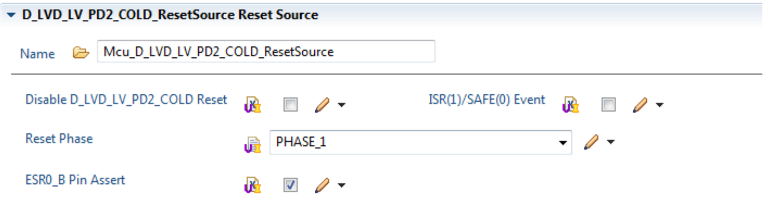
Property	Value
Label	Mcu Destructive Reset Threshold
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	15

4.9.16.4 Form Mcu_D_LVD_LV_PD2_COLD_ResetSource

LVD_LV_PD2_COLD reset configuration.

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_LVD_LV_PD2_COLD_ResetSource Reset Source

Name: Mcu_D_LVD_LV_PD2_COLD_ResetSource

Disable D_LVD_LV_PD2_COLD Reset □

ISR(1)/SAFE(0) Event □

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-20. Tresos Plugin snapshot for Mcu_D_LVD_LV_PD2_COLD_ResetSource form.

4.9.16.4.1 McuDisableReset (Mcu_D_LVD_LV_PD2_COLD_ResetSource)

RGM_FERD[D_LVD_LV_PD2_COLD] field configuration.

Disable temperature sensor destructive reset.

0 - A LVD LV PD2 COLD reset event generates a reset.

1 - A LVD LV PD2 COLD reset event generates a SAFE mode or an interrupt request.

Note: Implementation Specific Parameter.

Table 4-178. Attribute McuDisableReset (Mcu_D_LVD_LV_PD2_COLD_ResetSource) detailed description

Property	Value
Label	Disable D_LVD_LV_PD2_COLD Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.4.2 McuEnableInterruptSafe (Mcu_D_LVD_LV_PD2_COLD_ResetSource)

RGM_FEAR[AR_LVD_LV_PD2_COLD] field configuration.

Alternate Request for temperature sensor destructive reset.

0 - Generate a SAFE mode request.

1 - Generate an interrupt request.

Note: Implementation Specific Parameter.

Table 4-179. Attribute McuEnableInterruptSafe (Mcu_D_LVD_LV_PD2_COLD_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.4.3 McuResetPhase (Mcu_D_LVD_LV_PD2_COLD_ResetSource)

RGM_FESS[SS_LVD_LV_PD2_cold] field configuration. Short Sequence for voltage out of range functional reset. 0 - The reset sequence triggered by a voltage out of range functional reset event will start from PHASE1. 1 - The reset sequence triggered by a voltage out of range functional reset event will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

Table 4-180. Attribute McuResetPhase (Mcu_D_LVD_LV_PD2_COLD_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.4.4 McuResetPin (Mcu_D_LVD_LV_PD2_COLD_ResetSource)

RGM_FBRE[BE_LVD_LV_PD2_cold] field configuration. Bidirectional Reset Enable for PD2 LVD at Cold Point. 0 - ESR0_B is asserted on a PD2 LVD at cold point event if the reset is enabled. 1 - ESR0_B is not asserted on a PD2 LVD at cold point event. Note: Implementation Specific Parameter.

Table 4-181. Attribute McuResetPin (Mcu_D_LVD_LV_PD2_COLD_ResetSource) detailed description

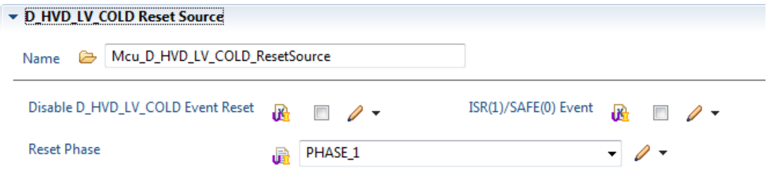
Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.5 Form Mcu_D_HVD_LV_COLD_ResetSource

HVD_LV_COLD reset configuration.

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_HVD_LV_COLD Reset Source

Name: Mcu_D_HVD_LV_COLD_ResetSource

Disable D_HVD_LV_COLD Event Reset ☐

ISR(1)/SAFE(0) Event ☐

Reset Phase: PHASE_1

Figure 4-21. Tresos Plugin snapshot for Mcu_D_HVD_LV_COLD_ResetSource form.

4.9.16.5.1 McuDisableReset (Mcu_D_HVD_LV_COLD_ResetSource)

RGM_FERD[D_HVD_LV_COLD] field configuration. Disable voltage out of range functional reset. 0 - A High Voltage Detect event triggers a reset. 1 - A High Voltage Detect event generates either a SAFE mode or an interrupt request depending on the value of RGM_FEAR[D_HVD_LV_COLD]. Note: Implementation Specific Parameter.

Table 4-182. Attribute McuDisableReset (Mcu_D_HVD_LV_COLD_ResetSource) detailed description

Property	Value
Label	Disable D_HVD_LV_COLD Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.5.2 McuEnableInterruptSafe (Mcu_D_HVD_LV_COLD_ResetSource)

RGM_FEAR[AR_HVD_LV_COLD] field configuration. Alternate Request for HVD_LV_COLD reset. 0 - Generate a SAFE mode request on a High Voltage Detect event if the reset is disabled. 1 - Generate an interrupt request on a High Voltage Detect event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-183. Attribute McuEnableInterruptSafe (Mcu_D_HVD_LV_COLD_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.5.3 McuResetPhase (Mcu_D_HVD_LV_COLD_ResetSource)

RGM_FESS[SS_HVD_LV_cold] field configuration. Short Sequence for voltage out of range functional reset. 0 - The reset sequence triggered by a voltage out of range functional reset event will start from PHASE1. 1 - The reset sequence triggered by a voltage out of range functional reset event will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

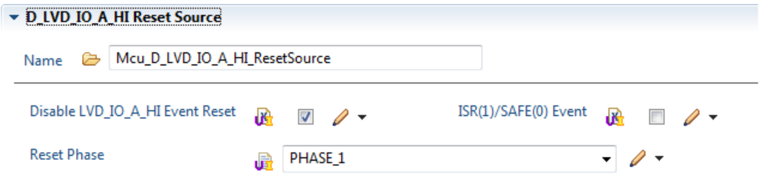
Table 4-184. Attribute McuResetPhase (Mcu_D_HVD_LV_COLD_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.6 Form Mcu_D_LVD_IO_A_HI_ResetSource

Temperature sensor functional reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_LVD_IO_A_HI Reset Source

Name Mcu_D_LVD_IO_A_HI_ResetSource

Disable LVD_IO_A_HI Event Reset ☑

ISR(1)/SAFE(0) Event □

Reset Phase PHASE_1

Figure 4-22. Tresos Plugin snapshot for Mcu_D_LVD_IO_A_HI_ResetSource form.

4.9.16.6.1 McuDisableReset (Mcu_D_LVD_IO_A_HI_ResetSource)

RGM_FERD[D_LVD_IO_A_HI] field register value. Disable LVD IO A HI reset. 0 - A LVD IO A HI reset event generates a reset. 1 - A LVD IO A HI reset event generates a SAFE mode or an interrupt request. Note: Implementation Specific Parameter.

Table 4-185. Attribute McuDisableReset (Mcu_D_LVD_IO_A_HI_ResetSource) detailed description

Property	Value
Label	Disable LVD_IO_A_HI Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.6.2 McuEnableInterruptSafe (Mcu_D_LVD_IO_A_HI_ResetSource)

RGM_FEAR[AR_LVD_IO_A_HI] field register value.

Alternate Request for temperature sensor functional reset.

0 - Generate a SAFE mode request on a LVD IO A HI event if the reset is disabled.

1 - Generate an interrupt request on a LVD IO A HI event if the reset is disabled.

Note: Implementation Specific Parameter.

Table 4-186. Attribute McuEnableInterruptSafe (Mcu_D_LVD_IO_A_HI_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom

Table continues on the next page...

Table 4-186. Attribute McuEnableInterruptSafe (Mcu_D_LVD_IO_A_HI_ResetSource) detailed description (continued)

Property	Value
Symbolic Name	false
Default	false

4.9.16.6.3 McuResetPhase (Mcu_D_LVD_IO_A_HI_ResetSource)

RGM_FESS[SS_LVD_IO_A_HI] field register value.

Short Sequence for temperature sensor functional reset.

0 - The reset sequence triggered by a LVD IO A HI reset event will start from PHASE1.

1 - The reset sequence triggered by a LVD IO A HI reset event will start from PHASE3, skipping PHASE1 and PHASE2.

Note: Implementation Specific Parameter.

Table 4-187. Attribute McuResetPhase (Mcu_D_LVD_IO_A_HI_ResetSource) detailed description

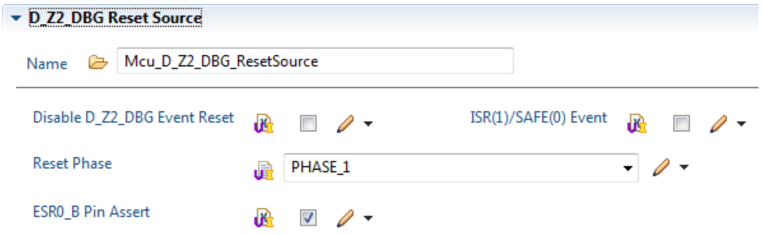
Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.7 Form Mcu_D_Z2_DBG_ResetSource

Z2 debug reset

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_Z2_DBG Reset Source

Name: Mcu_D_Z2_DBG_ResetSource

Disable D_Z2_DBG Event Reset □

ISR(1)/SAFE(0) Event □

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-23. Tresos Plugin snapshot for Mcu_D_Z2_DBG_ResetSource form.

4.9.16.7.1 McuDisableReset (Mcu_D_Z2_DBG_ResetSource)

RGM_FERD[D_Z2_DBG] field register value. Disable Z2 Debug 0 - A Z2 Debug event generates a reset. 1 - A Z2 Debug event generates a SAFE mode or an interrupt request. Note: Implementation Specific Parameter.

Table 4-188. Attribute McuDisableReset (Mcu_D_Z2_DBG_ResetSource) detailed description

Property	Value
Label	Disable D_Z2_DBG Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.7.2 McuEnableInterruptSafe (Mcu_D_Z2_DBG_ResetSource)

RGM_FEAR[AR_Z2_DBG] field configuration. Alternate Request for Z2_DBG reset. 0 - Generate a SAFE mode request on a Z2 Debug event if the reset is disabled. 1 - Generate an interrupt request on a Z2 Debug event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-189. Attribute McuEnableInterruptSafe (Mcu_D_Z2_DBG_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.7.3 McuResetPhase (Mcu_D_Z2_DBG_ResetSource)

RGM_FESS[SS_Z2_DBG] field register value.

Short Sequence for Z2 Debug Reset

0 - The reset sequence triggered by a Z2 Debug Reset event will start from PHASE1.

1 - The reset sequence triggered by a Z2 Debug Reset event will start from PHASE3, skipping PHASE1 and PHASE2.

Note: Implementation Specific Parameter.

Table 4-190. Attribute McuResetPhase (Mcu_D_Z2_DBG_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.7.4 McuResetPin (Mcu_D_Z2_DBG_ResetSource)

RGM_FBRE[BE_Z2_DBG] field configuration. Bidirectional Reset Enable for Z2 debug reset. 0 - ESR0_B is asserted on a Z2 debug reset event if the reset is enabled. 1 - ESR0_B is not asserted on a Z2 debug reset event. Note: Implementation Specific Parameter.

Table 4-191. Attribute McuResetPin (Mcu_D_Z2_DBG_ResetSource) detailed description

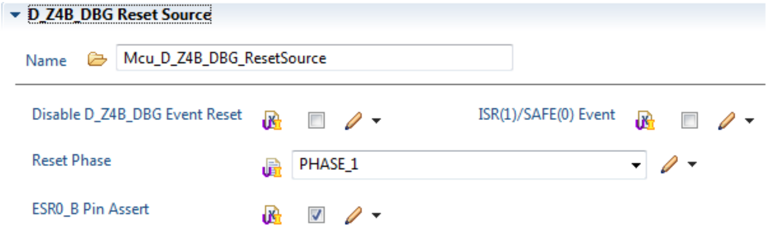
Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.8 Form Mcu_D_Z4B_DBG_ResetSource

Z4B debug reset

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_Z4B_DBG Reset Source

Name: Mcu_D_Z4B_DBG_ResetSource

Disable D_Z4B_DBG Event Reset ☐

ISR(1)/SAFE(0) Event ☐

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-24. Tresos Plugin snapshot for Mcu_D_Z4B_DBG_ResetSource form.

4.9.16.8.1 McuDisableReset (Mcu_D_Z4B_DBG_ResetSource)

RGM_FERD[D_Z4B_DBG] field register value. Disable Z4B Debug 0 - A Z4B Debug event generates a reset. 1 - A Z4B Debug event generates a SAFE mode or an interrupt request. Note: Implementation Specific Parameter.

Table 4-192. Attribute McuDisableReset (Mcu_D_Z4B_DBG_ResetSource) detailed description

Property	Value
Label	Disable D_Z4B_DBG Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.8.2 McuEnableInterruptSafe (Mcu_D_Z4B_DBG_ResetSource)

RGM_FEAR[AR_Z4B_DBG] field configuration. Alternate Request for Z4B_DBG reset. 0 - Generate a SAFE mode request on a Z4B Debug event if the reset is disabled. 1 - Generate an interrupt request on a Z4B Debug event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-193. Attribute McuEnableInterruptSafe (Mcu_D_Z4B_DBG_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.8.3 McuResetPhase (Mcu_D_Z4B_DBG_ResetSource)

RGM_FESS[SS_Z4B_DBG] field register value.

Short Sequence for Z4B Debug Reset

0 - The reset sequence triggered by a Z4B Debug Reset event will start from PHASE1.

1 - The reset sequence triggered by a Z4B Debug Reset event will start from PHASE3, skipping PHASE1 and PHASE2.

Note: Implementation Specific Parameter.

Table 4-194. Attribute McuResetPhase (Mcu_D_Z4B_DBG_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.8.4 McuResetPin (Mcu_D_Z4B_DBG_ResetSource)

RGM_FBRE[BE_Z4B_DBG] field configuration. Bidirectional Reset Enable for Z4B debug reset. 0 - ESR0_B is asserted on a Z4B debug reset event if the reset is enabled. 1 - ESR0_B is not asserted on a Z4B debug reset event. Note: Implementation Specific Parameter.

Table 4-195. Attribute McuResetPin (Mcu_D_Z4B_DBG_ResetSource) detailed description

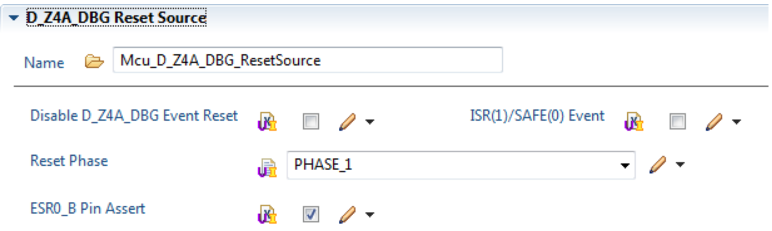
Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.9 Form Mcu_D_Z4A_DBG_ResetSource

Z4A debug reset

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_Z4A_DBG Reset Source

Name: Mcu_D_Z4A_DBG_ResetSource

Disable D_Z4A_DBG Event Reset □

ISR(1)/SAFE(0) Event □

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-25. Tresos Plugin snapshot for Mcu_D_Z4A_DBG_ResetSource form.

4.9.16.9.1 McuDisableReset (Mcu_D_Z4A_DBG_ResetSource)

RGM_FERD[D_Z4A_DBG] field register value. Disable Z4A Debug 0 - A Z4A Debug event generates a reset. 1 - A Z4A Debug event generates a SAFE mode or an interrupt request. Note: Implementation Specific Parameter.

Table 4-196. Attribute McuDisableReset (Mcu_D_Z4A_DBG_ResetSource) detailed description

Property	Value
Label	Disable D_Z4A_DBG Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.9.2 McuEnableInterruptSafe (Mcu_D_Z4A_DBG_ResetSource)

RGM_FEAR[AR_Z4A_DBG] field configuration. Alternate Request for Z4A_DBG reset. 0 - Generate a SAFE mode request on a Z4A Debug event if the reset is disabled. 1 - Generate an interrupt request on a Z4A Debug event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-197. Attribute McuEnableInterruptSafe (Mcu_D_Z4A_DBG_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.9.3 McuResetPhase (Mcu_D_Z4A_DBG_ResetSource)

RGM_FESS[SS_Z4A_DBG] field register value.

Short Sequence for Z4A Debug Reset

0 - The reset sequence triggered by a Z4A Debug Reset event will start from PHASE1.

1 - The reset sequence triggered by a Z4A Debug Reset event will start from PHASE3, skipping PHASE1 and PHASE2.

Note: Implementation Specific Parameter.

Table 4-198. Attribute McuResetPhase (Mcu_D_Z4A_DBG_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.9.4 McuResetPin (Mcu_D_Z4A_DBG_ResetSource)

RGM_FBRE[BE_Z4A_DBG] field configuration. Bidirectional Reset Enable for Z4A debug reset. 0 - ESR0_B is asserted on a Z4A debug reset event if the reset is enabled. 1 - ESR0_B is not asserted on a Z4A debug reset event. Note: Implementation Specific Parameter.

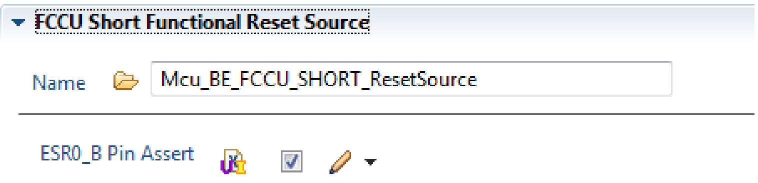
Table 4-199. Attribute McuResetPin (Mcu_D_Z4A_DBG_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.10 Form Mcu_BE_FCCU_SHORT_ResetSource

FCCU Short Functional Reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



FCCU Short Functional Reset Source

Name Mcu_BE_FCCU_SHORT_ResetSource

ESR0_B Pin Assert ☑

Figure 4-26. Tresos Plugin snapshot for Mcu_BE_FCCU_SHORT_ResetSource form.

4.9.16.10.1 McuResetPin (Mcu_BE_FCCU_SHORT_ResetSource)

RGM_FBRE[BE_FCCU_SHORT] field configuration. Bidirectional Reset Enable for FCCU Short Functional Reset. 0 - ESR0_B is asserted on a FCCU short functional reset event if the reset is enabled. 1 - ESR0_B is not asserted on a FCCU Short Functional Reset event. Note: Implementation Specific Parameter.

Table 4-200. Attribute McuResetPin (Mcu_BE_FCCU_SHORT_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN

Table continues on the next page...

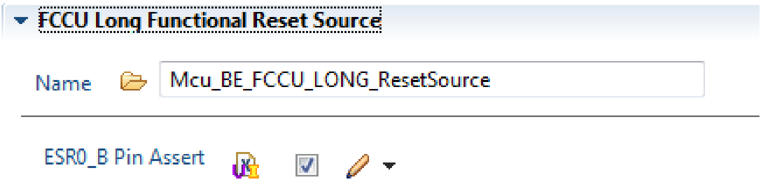
Table 4-200. Attribute McuResetPin (Mcu_BE_FCCU_SHORT_ResetSource) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.11 Form Mcu_BE_FCCU_LONG_ResetSource

FCCU Long Functional Reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



FCCU Long Functional Reset Source

Name: Mcu_BE_FCCU_LONG_ResetSource

ESR0_B Pin Assert ☑

Figure 4-27. Tresos Plugin snapshot for Mcu_BE_FCCU_LONG_ResetSource form.

4.9.16.11.1 McuResetPin (Mcu_BE_FCCU_LONG_ResetSource)

RGM_FBRE[BE_FCCU_LONG] field configuration. Bidirectional Reset Enable for FCCU Long Functional Reset. 0 - ESR0_B is asserted on a FCCU long functional reset event if the reset is enabled. 1 - ESR0_B is not asserted on a FCCU Long Functional Reset event. Note: Implementation Specific Parameter.

Table 4-201. Attribute McuResetPin (Mcu_BE_FCCU_LONG_ResetSource) detailed description

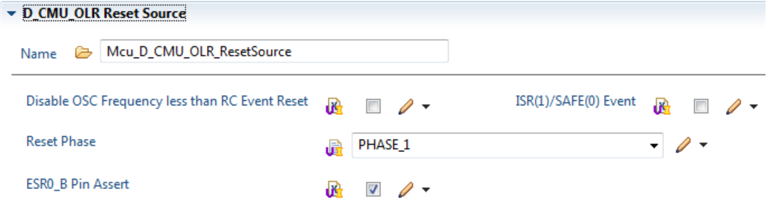
Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.12 Form Mcu_D_CMU_OLR_ResetSource

OSC Frequency less than RC reset

Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



D_CMU_OLR Reset Source

Name: Mcu_D_CMU_OLR_ResetSource

Disable OSC Frequency less than RC Event Reset □

ISR(1)/SAFE(0) Event □

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-28. Tresos Plugin snapshot for Mcu_D_CMU_OLR_ResetSource form.

4.9.16.12.1 McuDisableReset (Mcu_D_CMU_OLR_ResetSource)

RGM_FERD[D_CMU_OLR] field register value. Disable Z4A Debug 0 - OSC Frequency less than RC event triggers a reset sequence. 1 - OSC Frequency less than RC event generates either a SAFE mode or an interrupt request. Note: Implementation Specific Parameter.

Table 4-202. Attribute McuDisableReset (Mcu_D_CMU_OLR_ResetSource) detailed description

Property	Value
Label	Disable OSC Frequency less than RC Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.12.2 McuEnableInterruptSafe (Mcu_D_CMU_OLR_ResetSource)

RGM_FEAR[AR_CMU_OLR] field configuration. Alternate Request for OSC Frequency less than RC reset. 0 - Generate a SAFE mode request on a OSC Frequency less than RC event if the reset is disabled. 1 - Generate an interrupt request on a OSC Frequency less than RC event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-203. Attribute McuEnableInterruptSafe (Mcu_D_CMU_OLR_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.12.3 McuResetPhase (Mcu_D_CMU_OLR_ResetSource)

RGM_FESS[SS_CMU_OLR] field register value.

Short Sequence for OSC Frequency less than RC

0 - The reset sequence triggered by a OSC Frequency less than RC event will start from PHASE1.

1 - The reset sequence triggered by a OSC Frequency less than RC event will start from PHASE3, skipping PHASE1 and PHASE2.

Note: Implementation Specific Parameter.

Table 4-204. Attribute McuResetPhase (Mcu_D_CMU_OLR_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.12.4 McuResetPin (Mcu_D_CMU_OLR_ResetSource)

RGM_FBRE[BE_CMU_OLR] field configuration. Bidirectional Reset Enable for OSC Frequency less than RC. 0 - ESR0_B is asserted on a OSC Frequency less than RC event if the reset is enabled. 1 - ESR0_B is not asserted on a OSC Frequency less than RC event. Note: Implementation Specific Parameter.

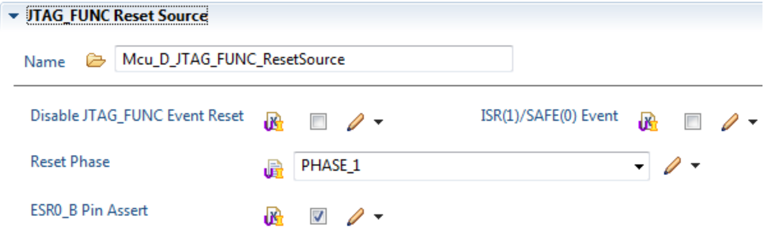
Table 4-205. Attribute McuResetPin (Mcu_D_CMU_OLR_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.13 Form Mcu_D_JTAG_FUNC_ResetSource

Disable JTAG Functional reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



JTAG_FUNC Reset Source

Name: Mcu_D_JTAG_FUNC_ResetSource

Disable JTAG_FUNC Event Reset □

ISR(1)/SAFE(0) Event □

Reset Phase: PHASE_1

ESR0_B Pin Assert ☑

Figure 4-29. Tresos Plugin snapshot for Mcu_D_JTAG_FUNC_ResetSource form.

4.9.16.13.1 McuDisableReset (Mcu_D_JTAG_FUNC_ResetSource)

RGM_FERD[D_JTAG_FUNC] field register value. Disable JTAG functional reset. 0 - A JTAG functional reset event triggers a reset sequence. 1 - A JTAG functional reset event generates either a SAFE mode or an interrupt request depending on the value of RGM_FEAR.AR_JTAG_FUNC. Note: Implementation Specific Parameter.

Table 4-206. Attribute McuDisableReset (Mcu_D_JTAG_FUNC_ResetSource) detailed description

Property	Value
Label	Disable JTAG_FUNC Event Reset
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.13.2 McuEnableInterruptSafe (Mcu_D_JTAG_FUNC_ResetSource)

RGM_FEAR[AR_JTAG_FUNC] field register value. Alternate Request for JTAG functional reset. 0 - Generate a SAFE mode request on a JTAG functional reset event if the reset is disabled. 1 - Generate an interrupt request on a JTAG functional reset event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-207. Attribute McuEnableInterruptSafe (Mcu_D_JTAG_FUNC_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.13.3 McuResetPhase (Mcu_D_JTAG_FUNC_ResetSource)

RGM_FESS[SS_JTAG_FUNC] field register value. Short Sequence for JTAG functional reset. 0 - The reset sequence triggered by a JTAG functional reset event will start from PHASE1. 1 - The reset sequence triggered by a JTAG functional reset event will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

Table 4-208. Attribute McuResetPhase (Mcu_D_JTAG_FUNC_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.13.4 McuResetPin (Mcu_D_JTAG_FUNC_ResetSource)

RGM_FBRE[BE_JTAG_FUNC] field configuration. Bidirectional Reset Enable for JTAG Function. 0 - ESR0_B is asserted on a JTAG Function event if the reset is enabled. 1 - ESR0_B is not asserted on a JTAG Function event. Note: Implementation Specific Parameter.

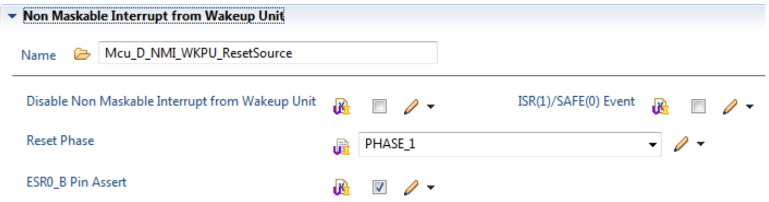
Table 4-209. Attribute McuResetPin (Mcu_D_JTAG_FUNC_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.14 Form Mcu_D_NMI_WKPU_ResetSource

Non Maskable Interrupt from Wakeup Unit. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



Non Maskable Interrupt from Wakeup Unit

Name Mcu_D_NMI_WKPU_ResetSource

Disable Non Maskable Interrupt from Wakeup Unit ☐

ISR(1)/SAFE(0) Event ☐

Reset Phase PHASE_1

ESR0_B Pin Assert ☑

Figure 4-30. Tresos Plugin snapshot for Mcu_D_NMI_WKPU_ResetSource form.

4.9.16.14.1 McuDisableReset (Mcu_D_NMI_WKPU_ResetSource)

RGM_FERD[D_NMI_WKPU] field register value. Disable Non Maskable Interrupt from Wakeup Unit. 0 - A Non Maskable Interrupt from Wakeup Unit event triggers a reset sequence. 1 - A Non Maskable Interrupt from Wakeup Unit event generates either a SAFE mode or an interrupt request depending on the value of RGM_FEAR.AR_JTAG_FUNC. Note: Implementation Specific Parameter.

Table 4-210. Attribute McuDisableReset (Mcu_D_NMI_WKPU_ResetSource) detailed description

Property	Value
Label	Disable Non Maskable Interrupt from Wakeup Unit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.14.2 McuEnableInterruptSafe (Mcu_D_NMI_WKPU_ResetSource)

RGM_FEAR[AR_NMI_WKPU] field register value. Alternate Request for Non Maskable Interrupt from Wakeup Unit. 0 - Generate a SAFE mode request on a Non Maskable Interrupt from Wakeup Unit event if the reset is disabled. 1 - Generate an interrupt request on a Non Maskable Interrupt from Wakeup Unit event if the reset is disabled. Note: Implementation Specific Parameter.

Table 4-211. Attribute McuEnableInterruptSafe (Mcu_D_NMI_WKPU_ResetSource) detailed description

Property	Value
Label	ISR(1)/SAFE(0) Event
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.16.14.3 McuResetPhase (Mcu_D_NMI_WKPU_ResetSource)

RGM_FESS[SS_NMI_WKPU] field register value. Short Sequence for Non Maskable Interrupt 3 Wakeup Unit. 0 - The reset sequence will start from PHASE1. 1 - The reset sequence will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

Table 4-212. Attribute McuResetPhase (Mcu_D_NMI_WKPU_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.14.4 McuResetPin (Mcu_D_NMI_WKPU_ResetSource)

RGM_FBRE[BE_NMI_WKPU] field configuration. Bidirectional Reset Enable for JTAG Function. 0 - ESR0_B is asserted on Non Maskable Interrupt from Wakeup Unit event if the reset is enabled. 1 - ESR0_B is not asserted on Non Maskable Interrupt from Wakeup Unit event. Note: Implementation Specific Parameter.

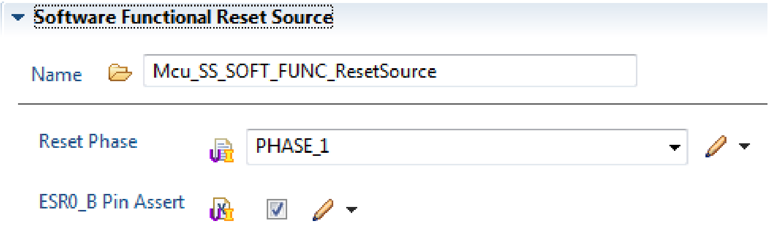
Table 4-213. Attribute McuResetPin (Mcu_D_NMI_WKPU_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.15 Form Mcu_SS_SOFT_FUNC_ResetSource

Software Functional reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



Software Functional Reset Source

Name: Mcu_SS_SOFT_FUNC_ResetSource

Reset Phase: PHASE_1

ESR0_B Pin Assert: ☑

Figure 4-31. Tresos Plugin snapshot for Mcu_SS_SOFT_FUNC_ResetSource form.

4.9.16.15.1 McuResetPhase (Mcu_SS_SOFT_FUNC_ResetSource)

RGM_FESS[SS_SOFT_FUNC] field register value. Short Sequence for SWT2 timeout reset. 0 - The reset sequence will start from PHASE1. 1 - The reset sequence will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

Table 4-214. Attribute McuResetPhase (Mcu_SS_SOFT_FUNC_ResetSource) detailed description

Property	Value
Label	Reset Phase
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

4.9.16.15.2 McuResetPin (Mcu_SS_SOFT_FUNC_ResetSource)

RGM_FBRE[BE_SOFT_FUNC] field configuration. Bidirectional Reset Enable for Software Generated Functional Reset. 0 - ESR0_B is asserted on a Software generated destructive reset event if the reset is enabled. 1 - ESR0_B is not asserted on a Software generated destructive reset event. Note: Implementation Specific Parameter.

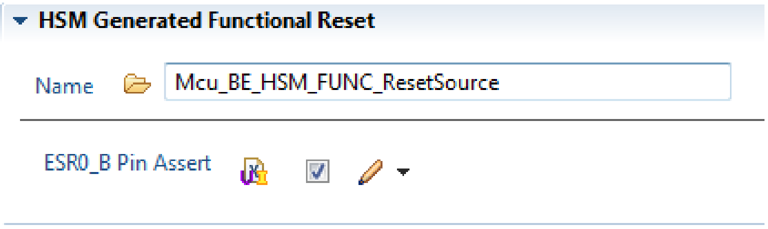
Table 4-215. Attribute McuResetPin (Mcu_SS_SOFT_FUNC_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.16 Form Mcu_BE_HSM_FUNC_ResetSource

HSM Generated Functional Reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



Name: Mcu_BE_HSM_FUNC_ResetSource

ESR0_B Pin Assert ☑

Figure 4-32. Tresos Plugin snapshot for Mcu_BE_HSM_FUNC_ResetSource form.

4.9.16.16.1 McuResetPin (Mcu_BE_HSM_FUNC_ResetSource)

RGM_FBRE[BE_HSM_FUNC] field configuration. Bidirectional Reset Enable for HSM Generated Functional Reset. 0 - ESR0_B is asserted on a HSM genrated functional reset event if the reset is enabled. 1 - ESR0_B is not asserted on a HSM genrated functional reset event. Note: Implementation Specific Parameter.

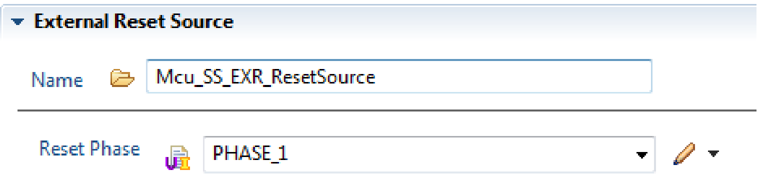
Table 4-216. Attribute McuResetPin (Mcu_BE_HSM_FUNC_ResetSource) detailed description

Property	Value
Label	ESR0_B Pin Assert
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.16.17 Form Mcu_SS_EXR_ResetSource

External Reset. Note: Implementation Specific Parameter.

Is included by form : Form McuResetConfig



Name: Mcu_SS_EXR_ResetSource

Reset Phase: PHASE_1

Figure 4-33. Tresos Plugin snapshot for Mcu_SS_EXR_ResetSource form.

4.9.16.17.1 McuResetPhase (Mcu_SS_EXR_ResetSource)

RGM_FESS[SS_EXR] field register value. Short Sequence for External Reset. 0 - The reset sequence will start from PHASE1. 1 - The reset sequence will start from PHASE3, skipping PHASE1 and PHASE2. Note: Implementation Specific Parameter.

Table 4-217. Attribute McuResetPhase (Mcu_SS_EXR_ResetSource) detailed description

Property	Value
Label	Reset Phase

Table continues on the next page...

Table 4-217. Attribute McuResetPhase (Mcu_SS_EXR_ResetSource) detailed description (continued)

Property	Value
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	PHASE_1
Range	PHASE_1 PHASE_3

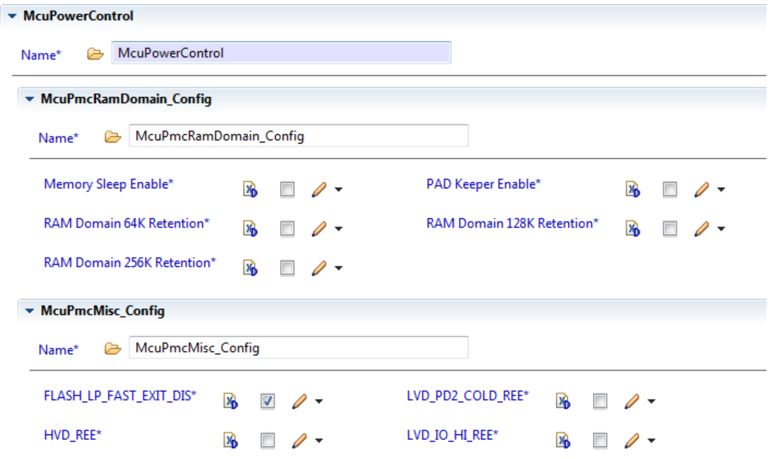
4.9.17 Form McuPowerControl

Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration

Included forms :

- Form McuPmcRamDomain_Config
- Form McuPmcMisc_Config



McuPowerControl

Name* McuPowerControl

McuPmcRamDomain_Config

Name* McuPmcRamDomain_Config

Memory Sleep Enable* □

PAD Keeper Enable* □

RAM Domain 64K Retention* □

RAM Domain 128K Retention* □

RAM Domain 256K Retention* □

McuPmcMisc_Config

Name* McuPmcMisc_Config

FLASH_LP_FAST_EXIT_DIS* ☑

LVD_PD2_COLD_REE* □

HVD_REE* □

LVD_IO_HI_REE* □

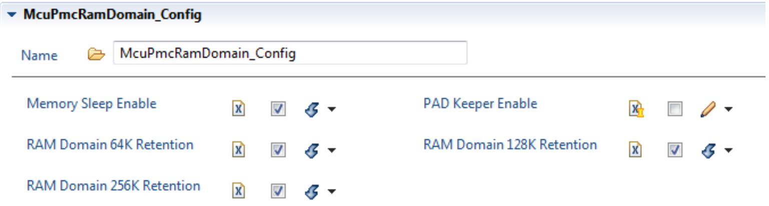
Figure 4-34. Tresos Plugin snapshot for McuPowerControl form.

4.9.17.1 Form McuPmcRamDomain_Config

RAM Domain Configuration.

Note: Implementation Specific Parameter.

Is included by form : Form McuPowerControl



McuPmcRamDomain_Config

Name: McuPmcRamDomain_Config

Memory Sleep Enable ☑

PAD Keeper Enable □

RAM Domain 64K Retention ☑

RAM Domain 128K Retention ☑

RAM Domain 256K Retention ☑

Figure 4-35. Tresos Plugin snapshot for McuPmcRamDomain_Config form.

4.9.17.1.1 McuPmcRdCr_MemSleep (McuPmcRamDomain_Config)

PMCDIG_RDCR[MEM_SLEEP_EN]

When set, will enable the Sleep state for memories. Program this bit in STOP/IOP_STOP to take advantage of memory sleep state.

0 - MEM_SLEEP disabled.

1 - MEM_SLEEP enabled.

Note: Implementation Specific Parameter.

Table 4-218. Attribute McuPmcRdCr_MemSleep (McuPmcRamDomain_Config) detailed description

Property	Value
Label	Memory Sleep Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.1.2 McuPmcRdCr_PadKeep (McuPmcRamDomain_Config)

PMCDIG_RDCR[PAD_KEEP_EN]

Enables the PAD keeping functionality where some Standby PADs can retain their configuration while in Power gated Modes (Standby).

0 - PAD Keeping functionality disabled.

1 - PAD Keeping functionality enabled.

Note: Implementation Specific Parameter.

Table 4-219. Attribute McuPmcRdCr_PadKeep (McuPmcRamDomain_Config) detailed description

Property	Value
Label	PAD Keeper Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.1.3 McuPmcRdCr_Rd64Ret (McuPmcRamDomain_Config)

PMCDIG_RDCR[RD64_RET].

Retains 64K RAM while in Standby Mode.

0 - Switch to RAM supplies Opened (RAM cut is shut off), only 8K are left.

1 - Switch to RAM supplies closed.

Note: Implementation Specific Parameter.

Table 4-220. Attribute McuPmcRdCr_Rd64Ret (McuPmcRamDomain_Config) detailed description

Property	Value
Label	RAM Domain 64K Retention
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.1.4 McuPmcRdCr_Rd128Ret (McuPmcRamDomain_Config)

PMCDIG_RDCR[RD128_RET].

Retains 128K RAM while in Standby Mode.

0 - Switch to RAM supplies Opened (RAM cut is shut off).

1 - Switch to RAM supplies closed.

Note: Implementation Specific Parameter.

Table 4-221. Attribute McuPmcRdCr_Rd128Ret (McuPmcRamDomain_Config) detailed description

Property	Value
Label	RAM Domain 128K Retention
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.1.5 McuPmcRdCr_Rd256Ret (McuPmcRamDomain_Config)

PMCDIG_RDCR[RD256_RET].

Retains 256K RAM while in Standby Mode.

0 - Switch to RAM supplies Opened (RAM cut is shut off)>

1 - Switch to RAM supplies closed.

Note: Implementation Specific Parameter.

Table 4-222. Attribute McuPmcRdCr_Rd256Ret (McuPmcRamDomain_Config) detailed description

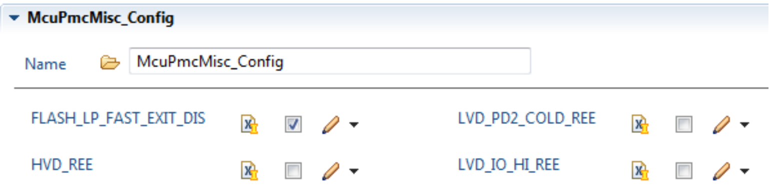
Property	Value
Label	RAM Domain 256K Retention
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.2 Form McuPmcMisc_Config

MISC Control Register.

Note: Implementation Specific Parameter.

Is included by form : Form McuPowerControl



McuPmcMisc_Config

Name McuPmcMisc_Config

FLASH_LP_FAST_EXIT_DIS ☑

LVD_PD2_COLD_REE □

HVD_REE □

LVD_IO_HI_REE □

Figure 4-36. Tresos Plugin snapshot for McuPmcMisc_Config form.

4.9.17.2.1 McuPmcMcr_FlashLpFastExistDis (McuPmcMisc_Config)

PMCDIG_MCR[FLASH_LP_FAST_EXIT_DIS]

Flash Low Power Fast Exit Disable

When set, the Flash will not be enabled for fast wakeup sequences. This results in increased wakeup times for Flash but in turn enables a Robust Wakeup sequence.

Note: Implementation Specific Parameter.

Table 4-223. Attribute McuPmcMcr_FlashLpFastExistDis (McuPmcMisc_Config) detailed description

Property	Value
Label	FLASH_LP_FAST_EXIT_DIS
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.2.2 McuPmcMcr_LvdPd2ColdRee (McuPmcMisc_Config)

PMCDIG_MCR[LVD_PD2_COLD_REE]

LVD_PD2 Cold Point Flag Reset Event Enable

Enables the LVD_PD2 Cold Point Flag functional reset source to RGM if set.

Note: Please ensure before entering STANDBY0 or LPU_RUN Mode, PMCDIG_MCR[LVD_PD2_COLD_REE] bit is programmed to 0 (Eraata ERR010447).

Note: Implementation Specific Parameter.

Table 4-224. Attribute McuPmcMcr_LvdPd2ColdRee (McuPmcMisc_Config) detailed description

Property	Value
Label	LVD_PD2_COLD_REE
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.2.3 McuPmcMcr_HvdRee (McuPmcMisc_Config)

PMCDIG_MCR[HVD_REE]

HVD Reset Event Enable

Enables the HVD functional reset source to RGM if set.

Note: Implementation Specific Parameter.

Table 4-225. Attribute McuPmcMcr_HvdRee (McuPmcMisc_Config) detailed description

Property	Value
Label	HVD_REE
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.17.2.4 McuPmcMcr_LvdIoHiRee (McuPmcMisc_Config)

PMCDIG_MCR[LVD_IO_HI_REE]

HVD Reset Event Enable

Enables the HVD functional reset source to RGM if set.

Note: Please ensure before entering STANDBY0 or LPU_RUN Mode, PMCDIG_MCR[LVD_IO_HI_REE] bit is programmed to 0 (Eraata ERR010447).

Note: Implementation Specific Parameter.

Table 4-226. Attribute McuPmcMcr_LvdIoHiRee (McuPmcMisc_Config) detailed description

Property	Value
Label	LVD_IO_HI_REE
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.18 Form McuClockSettingConfig

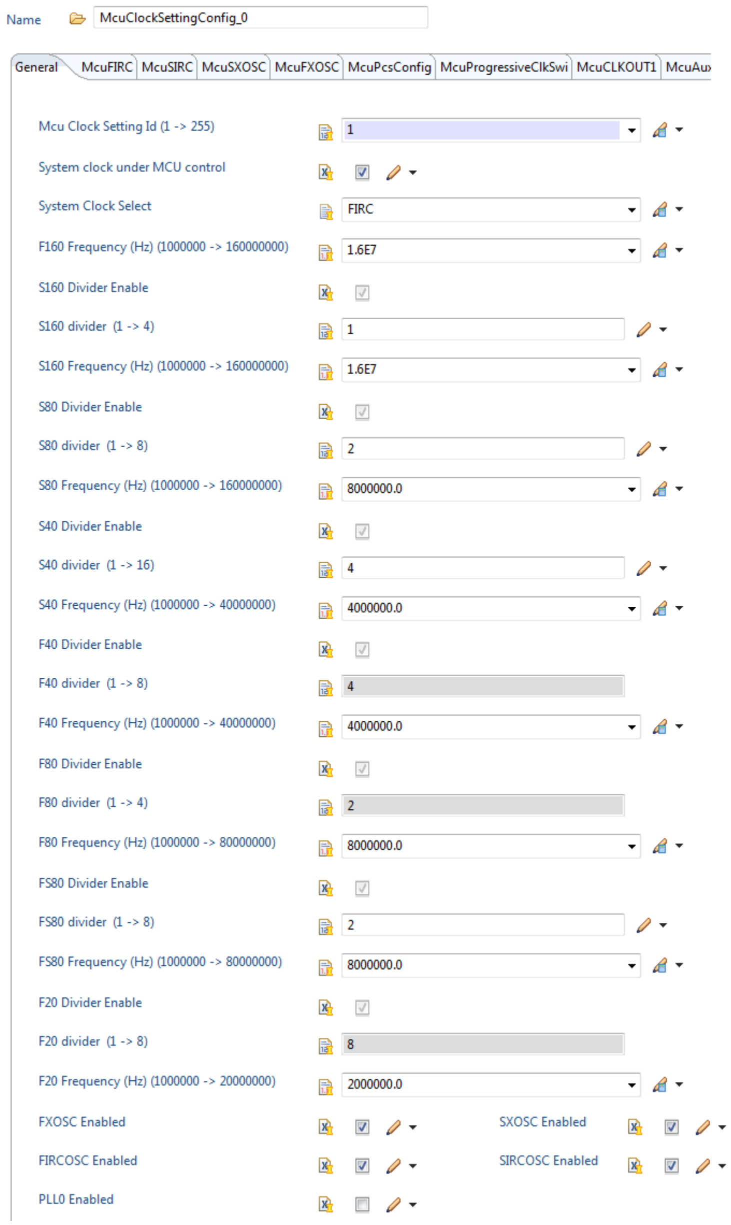
This container contains the configuration for the Clock settings of the MCU.

Is included by form : Form McuModuleConfiguration

Included forms :

- Form McuFIRC
- Form McuSIRC
- Form McuSXOSC
- Form McuFXOSC
- Form McuProgClkSwitch
- Form McuCLKOUT1
- Form McuAuxClock2
- Form McuAuxClock4
- Form McuAuxClock5
- Form McuAuxClock6
- Form McuAuxClock8
- Form McuAuxClock9
- Form McuPll_0
- Form McuEMIOS
- Form McuClkMonitor_0
- Form McuFlash
- Form McuRam

- Form McuClockReferencePoint



Name: McuClockSettingConfig_0

Mcu Clock Setting Id (1 -> 255)	1
System clock under MCU control	☑
System Clock Select	FIRC
F160 Frequency (Hz) (1000000 -> 160000000)	1.6E7
S160 Divider Enable	☑
S160 divider (1 -> 4)	1
S160 Frequency (Hz) (1000000 -> 160000000)	1.6E7
S80 Divider Enable	☑
S80 divider (1 -> 8)	2
S80 Frequency (Hz) (1000000 -> 160000000)	8000000.0
S40 Divider Enable	☑
S40 divider (1 -> 16)	4
S40 Frequency (Hz) (1000000 -> 40000000)	4000000.0
F40 Divider Enable	☑
F40 divider (1 -> 8)	4
F40 Frequency (Hz) (1000000 -> 40000000)	4000000.0
F80 Divider Enable	☑
F80 divider (1 -> 4)	2
F80 Frequency (Hz) (1000000 -> 80000000)	8000000.0
FS80 Divider Enable	☑
FS80 divider (1 -> 8)	2
FS80 Frequency (Hz) (1000000 -> 80000000)	8000000.0
F20 Divider Enable	☑
F20 divider (1 -> 8)	8
F20 Frequency (Hz) (1000000 -> 20000000)	2000000.0
FXOSC Enabled	☑
SXOSC Enabled	☑
FIRCOSC Enabled	☑
SIRCOSC Enabled	☑
PLL0 Enabled	□

Figure 4-37. Tresos Plugin snapshot for McuClockSettingConfig form.

4.9.18.1 McuClockSettingId (McuClockSettingConfig)

The Id of this McuClockSettingConfig to be used as argument for the API call Mcu_InitClock().

Table 4-227. Attribute McuClockSettingId (McuClockSettingConfig) detailed description

Property	Value
Label	Mcu Clock Setting Id
Type	INTEGER

Table continues on the next page...

Table 4-227. Attribute McuClockSettingId (McuClockSettingConfig) detailed description (continued)

Property	Value
Origin	AUTOSAR_ECUC
Symbolic Name	true
Invalid	Range <=255 >=0

4.9.18.2 McuClksysClockMcuControl (McuClockSettingConfig)

0 - System clock tree is NOT under mcu control.

1 - System clock is under mcu control.

If this is set to false, the MCU code will not configure the system clock registers when Mcu_InitClock is called

Note: Implementation Specific Parameter.

Table 4-228. Attribute McuClksysClockMcuControl (McuClockSettingConfig) detailed description

Property	Value
Label	System clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.3 McuSystemClockSwitch (McuClockSettingConfig)

System Clock Select. Configure the ME_mode_MC[SYSCLK] register field.

Note: Implementation Specific Parameter.

Table 4-229. Attribute McuSystemClockSwitch (McuClockSettingConfig) detailed description

Property	Value
Label	System Clock Select
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.4 McuSystemClockF160Freq (McuClockSettingConfig)

This is the frequency for the F160 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-230. Attribute McuSystemClockF160Freq (McuClockSettingConfig) detailed description

Property	Value
Label	F160 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.5 McuSystemClkDiv0_En (McuClockSettingConfig)

CGM_SC_DC0[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-231. Attribute McuSystemClkDiv0_En (McuClockSettingConfig) detailed description

Property	Value
Label	S160 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.6 McuSystemClkDiv_Divider0 (McuClockSettingConfig)

CGM_SC_DC0[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider0-1).

This field controlls the S160 clock.

Note: Implementation Specific Parameter.

Table 4-232. Attribute McuSystemClkDiv_Divider0 (McuClockSettingConfig) detailed description

Property	Value
Label	S160 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.7 McuSystemClockS160Freq (McuClockSettingConfig)

This is the frequency for the S160 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-233. Attribute McuSystemClockS160Freq (McuClockSettingConfig) detailed description

Property	Value
Label	S160 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.8 McuSystemClkDiv1_En (McuClockSettingConfig)

CGM_SC_DC1[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-234. Attribute McuSystemClkDiv1_En (McuClockSettingConfig) detailed description

Property	Value
Label	S80 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.9 McuSystemClkDiv_Divider1 (McuClockSettingConfig)

CGM_SC_DC1[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider1-1).

This field controlls the S80 clock.

Note: Implementation Specific Parameter.

Table 4-235. Attribute McuSystemClkDiv_Divider1 (McuClockSettingConfig) detailed description

Property	Value
Label	S80 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2

4.9.18.10 McuSystemClockS80Freq (McuClockSettingConfig)

This is the frequency for the S80 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-236. Attribute McuSystemClockS80Freq (McuClockSettingConfig) detailed description

Property	Value
Label	S80 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.11 McuSystemClkDiv2_En (McuClockSettingConfig)

CGM_SC_DC2[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-237. Attribute McuSystemClkDiv2_En (McuClockSettingConfig) detailed description

Property	Value
Label	S40 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.12 McuSystemClkDiv_Divider2 (McuClockSettingConfig)

CGM_SC_DC2[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider2-1).

This field controlls the S40 clock.

Note: Implementation Specific Parameter.

Table 4-238. Attribute McuSystemClkDiv_Divider2 (McuClockSettingConfig) detailed description

Property	Value
Label	S40 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4

4.9.18.13 McuSystemClockS40Freq (McuClockSettingConfig)

This is the frequency for the S40 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-239. Attribute McuSystemClockS40Freq (McuClockSettingConfig) detailed description

Property	Value
Label	S40 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.14 McuSystemClkDiv3_En (McuClockSettingConfig)

CGM_SC_DC3[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-240. Attribute McuSystemClkDiv3_En (McuClockSettingConfig) detailed description

Property	Value
Label	F40 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.15 McuSystemClkDiv_Divider3 (McuClockSettingConfig)

CGM_SC_DC3[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider3-1).

This field controlls the F40 clock.

Note: Implementation Specific Parameter.

Table 4-241. Attribute McuSystemClkDiv_Divider3 (McuClockSettingConfig) detailed description

Property	Value
Label	F40 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4

4.9.18.16 McuSystemClockF40Freq (McuClockSettingConfig)

This is the frequency for the F40 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-242. Attribute McuSystemClockF40Freq (McuClockSettingConfig) detailed description

Property	Value
Label	F40 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.17 McuSystemClkDiv4_En (McuClockSettingConfig)

CGM_SC_DC4[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-243. Attribute McuSystemClkDiv4_En (McuClockSettingConfig) detailed description

Property	Value
Label	F80 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.18 McuSystemClkDiv_Divider4 (McuClockSettingConfig)

CGM_SC_DC4[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider4-1).

This field controlls the F80 clock.

Note: Implementation Specific Parameter.

Table 4-244. Attribute McuSystemClkDiv_Divider4 (McuClockSettingConfig) detailed description

Property	Value
Label	F80 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2

4.9.18.19 McuSystemClockF80Freq (McuClockSettingConfig)

This is the frequency for the F80 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-245. Attribute McuSystemClockF80Freq (McuClockSettingConfig) detailed description

Property	Value
Label	F80 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.20 McuSystemClkDiv5_En (McuClockSettingConfig)

CGM_SC_DC5[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-246. Attribute McuSystemClkDiv5_En (McuClockSettingConfig) detailed description

Property	Value
Label	FS80 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.21 McuSystemClkDiv_Divider5 (McuClockSettingConfig)

CGM_SC_DC5[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider5-1).

This field controlls the FS80 clock.

Note: Implementation Specific Parameter.

Table 4-247. Attribute McuSystemClkDiv_Divider5 (McuClockSettingConfig) detailed description

Property	Value
Label	FS80 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2

4.9.18.22 McuSystemClockFS80Freq (McuClockSettingConfig)

This is the frequency for the FS80 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-248. Attribute McuSystemClockFS80Freq (McuClockSettingConfig) detailed description

Property	Value
Label	FS80 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.23 McuSystemClkDiv6_En (McuClockSettingConfig)

CGM_SC_DC6[DE] field register. System clock divider 0. 0 - Disable system clock divider 0. 1 - Enable system clock divider 0. Note: Implementation Specific Parameter.

Table 4-249. Attribute McuSystemClkDiv6_En (McuClockSettingConfig) detailed description

Property	Value
Label	F20 Divider Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.24 McuSystemClkDiv_Divider6 (McuClockSettingConfig)

CGM_SC_DC6[DIV] field register.

Register is configured with (McuSystemClkDiv_Divider6-1).

This field controlls the F20 clock.

Note: Implementation Specific Parameter.

Table 4-250. Attribute McuSystemClkDiv_Divider6 (McuClockSettingConfig) detailed description

Property	Value
Label	F20 divider
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	8

4.9.18.25 McuSystemClockF20Freq (McuClockSettingConfig)

This is the frequency for the F20 core clock.

Value calculated for user info. It is given in Hz.

Note: Implementation Specific Parameter.

Table 4-251. Attribute McuSystemClockF20Freq (McuClockSettingConfig) detailed description

Property	Value
Label	F20 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.26 McuClkSetFXOSC_En (McuClockSettingConfig)

0 - FXOSC is disabled in this mode. 1 - FXOSC is enabled in this mode and can be used as a clock source. Note: Implementation Specific Parameter.

Table 4-252. Attribute McuClkSetFXOSC_En (McuClockSettingConfig) detailed description

Property	Value
Label	FXOSC Enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.27 McuClkSetSXOSC_En (McuClockSettingConfig)

0 - SXOSC is disabled in this mode. 1 - SXOSC is enabled in this mode and can be used as a clock source. Note: Implementation Specific Parameter.

Table 4-253. Attribute McuClkSetSXOSC_En (McuClockSettingConfig) detailed description

Property	Value
Label	SXOSC Enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.28 McuClkSetFIRCOSC_En (McuClockSettingConfig)

0 - FIRCOSC is disabled in this mode. 1 - FIRCOSC is enabled in this mode and can be used as a clock source. Note: Implementation Specific Parameter.

Table 4-254. Attribute McuClkSetFIRCOSC_En (McuClockSettingConfig) detailed description

Property	Value
Label	FIRCOSC Enabled
Type	BOOLEAN

Table continues on the next page...

Table 4-254. Attribute McuClkSetFIRCOSC_En (McuClockSettingConfig) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.29 McuClkSetSIRCOSC_En (McuClockSettingConfig)

0 - SIRCOSC is disabled in this mode. 1 - SIRCOSC is enabled in this mode and can be used as a clock source. Note: Implementation Specific Parameter.

Table 4-255. Attribute McuClkSetSIRCOSC_En (McuClockSettingConfig) detailed description

Property	Value
Label	SIRCOSC Enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.30 McuClkSetPLL0_En (McuClockSettingConfig)

0 - PLL0 is disabled in this mode. 1 - PLL0 is enabled in this mode and can be used as a clock source. Note: Implementation Specific Parameter.

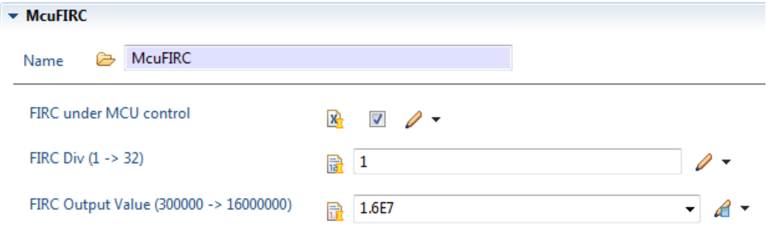
Table 4-256. Attribute McuClkSetPLL0_En (McuClockSettingConfig) detailed description

Property	Value
Label	PLL0 Enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.31 Form McuFIRC

This container contains the specific configuration of the FIRC Oscillator configuration.
Note: Implementation specific Container.

Is included by form : Form McuClockSettingConfig



McuFIRC

Name: McuFIRC

FIRC under MCU control ☑

FIRC Div (1 -> 32): 1

FIRC Output Value (300000 -> 16000000): 1.6E7

Figure 4-38. Tresos Plugin snapshot for McuFIRC form.

4.9.18.31.1 McuFIRCUnderMcuControl (McuFIRC)

Set this to TRUE if FIRC is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-257. Attribute McuFIRCUnderMcuControl (McuFIRC) detailed description

Property	Value
Label	FIRC under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.31.2 McuFIRC_Div (McuFIRC)

The FIRC Divider bits select the clock divider factor.

The FIRC_CTL[FIRCDIV] register is written with (FIRC Div) - 1.

Note: Implementation Specific Parameter.

Table 4-258. Attribute McuFIRC_Div (McuFIRC) detailed description

Property	Value
Label	FIRC Div
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.31.3 McuFIRC_DivOutputValue (McuFIRC)

The FIRC Output value, after the divider factor.

Note: Implementation Specific Parameter.

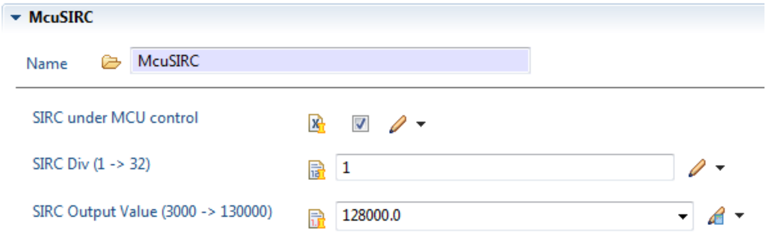
Table 4-259. Attribute McuFIRC_DivOutputValue (McuFIRC) detailed description

Property	Value
Label	FIRC Output Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.32 Form McuSIRC

This container contains the specific configuration of the SIRC Oscillator configuration. Note: Implementation specific Container.

Is included by form : Form McuClockSettingConfig



McuSIRC

Name: McuSIRC

SIRC under MCU control ☑

SIRC Div (1 -> 32): 1

SIRC Output Value (3000 -> 130000): 128000.0

Figure 4-39. Tresos Plugin snapshot for McuSIRC form.

4.9.18.32.1 McuSIRCUnderMcuControl (McuSIRC)

Set this to TRUE if SIRC is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-260. Attribute McuSIRCUnderMcuControl (McuSIRC) detailed description

Property	Value
Label	SIRC under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.32.2 McuSIRC_Div (McuSIRC)

The SIRC Divider bits select the clock divider factor.

The SIRC_CTL[SIRCDIV] register is written with (SIRC Div) - 1.

Note: Implementation Specific Parameter.

Table 4-261. Attribute McuSIRC_Div (McuSIRC) detailed description

Property	Value
Label	SIRC Div
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.32.3 McuSIRC_DivOutputValue (McuSIRC)

The SIRC Divider bits select the clock divider factor.

The SIRC_CTL[SIRCDIV] register is written with (SIRC Div) - 1.

Note: Implementation Specific Parameter.

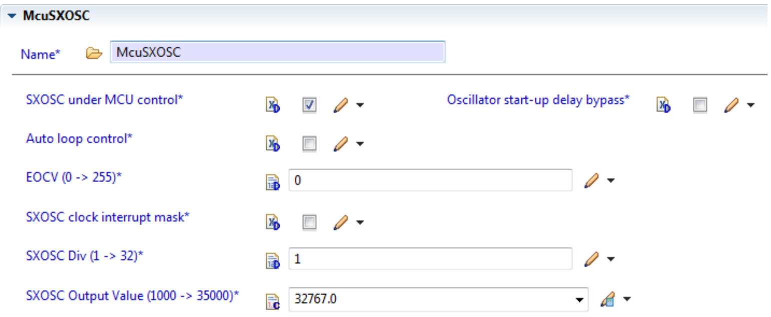
Table 4-262. Attribute McuSIRC_DivOutputValue (McuSIRC) detailed description

Property	Value
Label	SIRC Output Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.33 Form McuSXOSC

This container contains the specific configuration of the SXOSC Oscillator configuration. Note: Implementation specific Container.

Is included by form : Form McuClockSettingConfig



McuSXOSC

Name* McuSXOSC

SXOSC under MCU control* ☑

Oscillator start-up delay bypass* □

Auto loop control* □

EOCV (0 -> 255)* 0

SXOSC clock interrupt mask* □

SXOSC Div (1 -> 32)* 1

SXOSC Output Value (1000 -> 35000)* 32767.0

Figure 4-40. Tresos Plugin snapshot for McuSXOSC form.

4.9.18.33.1 McuSXOSCUnderMcuControl (McuSXOSC)

Set this to TRUE if SXOSC is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-263. Attribute McuSXOSCUnderMcuControl (McuSXOSC) detailed description

Property	Value
Label	SXOSC under MCU control
Type	BOOLEAN

Table continues on the next page...

Table 4-263. Attribute McuSXOSCUnderMcuControl (McuSXOSC) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.33.2 McuSXOSC_OSCBYP (McuSXOSC)

Oscillator start-up delay bypass - SXOSC_CTL[OSCBYP].

This bit specifies whether to bypass the EOCV counter, which provides startup stabilization time for the crystal oscillator. When this bit is set, the oscillator starts immediately when the SXOSC is selected, instead of waiting for the stabilization time to elapse. The user can only set this bit. System reset is needed to clear this bit.

Table 4-264. Detailed description.

0	The EOCV counter is not bypassed, and the oscillator starts after the stabilization period has elapsed.
1	The EOCV counter is bypassed, and the oscillator starts immediately.

Note: Implementation Specific Parameter.

Table 4-265. Attribute McuSXOSC_OSCBYP (McuSXOSC) detailed description

Property	Value
Label	Oscillator start-up delay bypass
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.33.3 McuSXOSC_ALC (McuSXOSC)

Auto level control - SXOSC_CTL[ALC].

The ALC bit controls the current provided to the crystal.

Table 4-266. Detailed description.

0	Auto level control disable
1	Auto level control enable

Note: Implementation Specific Parameter.

Table 4-267. Attribute McuSXOSC_ALC (McuSXOSC) detailed description

Property	Value
Label	Auto level control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.33.4 McuSXOSC_EOCV (McuSXOSC)

End of Counter Value - SXOSC_CTL[EOCV]

These bits specify the end of the count value to be used for comparison by the SXOSC stabilization counter

Note: Implementation Specific Parameter.

Table 4-268. Attribute McuSXOSC_EOCV (McuSXOSC) detailed description

Property	Value
Label	EOCV
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.33.5 McuSXOSC_M_OSC (McuSXOSC)

SXOSC clock interrupt mask - SXOSC_CTL[M_OSC].

Table 4-269. Detailed description.

0	SXOSC clock interrupt is masked
1	SXOSC clock interrupt is enabled

Note: Implementation Specific Parameter.

Table 4-270. Attribute McuSXOSC_M_OSC (McuSXOSC) detailed description

Property	Value
Label	SXOSC clock interrupt mask
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.33.6 McuSXOSC_Div (McuSXOSC)

The SXOSC Divider bits select the clock divider factor.

The SXOSC_CTL[OSCDIV] register is written with (SXOSC Div) - 1.

Note: Implementation Specific Parameter.

Table 4-271. Attribute McuSXOSC_Div (McuSXOSC) detailed description

Property	Value
Label	SXOSC Div
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.33.7 McuSXOSC_DivOutputValue (McuSXOSC)

The SXOSC Divider bits select the clock divider factor.

The SXOSC_CTL[SXOSC] register is written with (SXOSC Div) - 1.

Note: Implementation Specific Parameter.

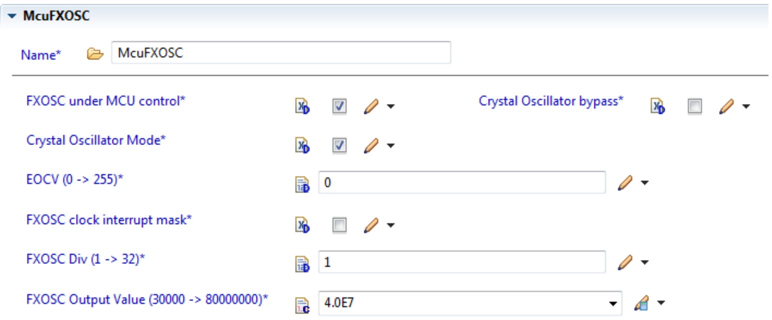
Table 4-272. Attribute McuSXOSC_DivOutputValue (McuSXOSC) detailed description

Property	Value
Label	SXOSC Output Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.34 Form McuFXOSC

This container contains the specific configuration of the FXOSC Oscillator configuration. Note: Implementation specific Container.

Is included by form : Form McuClockSettingConfig



McuFXOSC

Name* McuFXOSC

FXOSC under MCU control* ☑

Crystal Oscillator bypass* □

Crystal Oscillator Mode* ☑

EOCV (0 -> 255)* 0

FXOSC clock interrupt mask* □

FXOSC Div (1 -> 32)* 1

FXOSC Output Value (30000 -> 80000000)* 4.0E7

Figure 4-41. Tresos Plugin snapshot for McuFXOSC form.

4.9.18.34.1 McuFXOSCUnderMcuControl (McuFXOSC)

Set this to TRUE if FXOSC is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-273. Attribute McuFXOSCUnderMcuControl (McuFXOSC) detailed description

Property	Value
Label	FXOSC under MCU control
Type	BOOLEAN

Table continues on the next page...

Table 4-273. Attribute McuFXOSCUnderMcuControl (McuFXOSC) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.34.2 McuFXOSC_OSCBYP (McuFXOSC)

Crystal Oscillator bypass - FXOSC_CTL[OSCBYP].

Table 4-274. Detailed description.

0	Oscillator output is used as root clock.
1	EXTAL is used as root clock.

Note: Implementation Specific Parameter.

Table 4-275. Attribute McuFXOSC_OSCBYP (McuFXOSC) detailed description

Property	Value
Label	Crystal Oscillator bypass
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.34.3 McuFXOSC_OSCM (McuFXOSC)

Crystal Oscillator Mode - FXOSC_CTL[OSCM].

This bit selects the oscillator mode when not in bypass.

Table 4-276. Detailed description.

0(unchecked)	FSP mode (full swing pierce mode)
1(checked)	LCP mode (loop controlled pierce mode)

Note: Implementation Specific Parameter.

Table 4-277. Attribute McuFXOSC_OSCM (McuFXOSC) detailed description

Property	Value
Label	Crystal Oscillator Mode
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.34.4 McuFXOSC_EOCV (McuFXOSC)

End of Counter Value - FXOSC_CTL[EOCV]

These bits specify the end of the count value to be used for comparison by the SXOSC stabilization counter

Note: Implementation Specific Parameter.

Table 4-278. Attribute McuFXOSC_EOCV (McuFXOSC) detailed description

Property	Value
Label	EOCV
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.34.5 McuFXOSC_M_OSC (McuFXOSC)

SXOSC clock interrupt mask - FXOSC_CTL[M_OSC].

Table 4-279. Detailed description.

0	FXOSC clock interrupt is masked
1	FXOSC clock interrupt is enabled

Note: Implementation Specific Parameter.

Table 4-280. Attribute McuFXOSC_M_OSC (McuFXOSC) detailed description

Property	Value
Label	FXOSC clock interrupt mask
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.34.6 McuFXOSC_Div (McuFXOSC)

The FXOSC Divider bits select the clock divider factor.

The FXOSC_CTL[OSCDIV] register is written with (FXOSC Div) - 1.

Note: Implementation Specific Parameter.

Table 4-281. Attribute McuFXOSC_Div (McuFXOSC) detailed description

Property	Value
Label	FXOSC Div
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.34.7 McuFXOSC_DivOutputValue (McuFXOSC)

The FXOSC Divider bits select the clock divider factor.

The FXOSC_CTL[FXOSC] register is written with (FXOSC Div) - 1.

Note: Implementation Specific Parameter.

Table 4-282. Attribute McuFXOSC_DivOutputValue (McuFXOSC) detailed description

Property	Value
Label	FXOSC Output Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

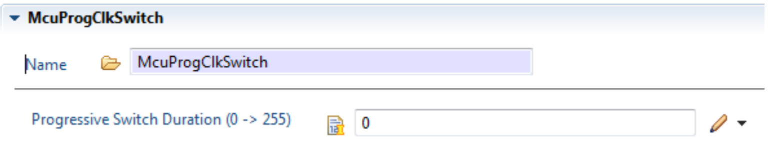
4.9.18.35 Form McuProgClkSwitch

This container provides the specific configuration for Progresisve Clock Switch. Note: Implementation Specific Container.

Is included by form : Form McuClockSettingConfig

Included forms :

- Form McuPcsConfig



McuProgClkSwitch

Name McuProgClkSwitch

Progressive Switch Duration (0 -> 255) 0

Figure 4-42. Tresos Plugin snapshot for McuProgClkSwitch form.

4.9.18.35.1 McuProgresSwitchDuration (McuProgClkSwitch)

CGM_PCS_SDUR register configuration. This value defines the duration of one PCS clock switch step in terms of 16 MHz int. RC osc. cycles. Note: Implementation Specific Parameter.

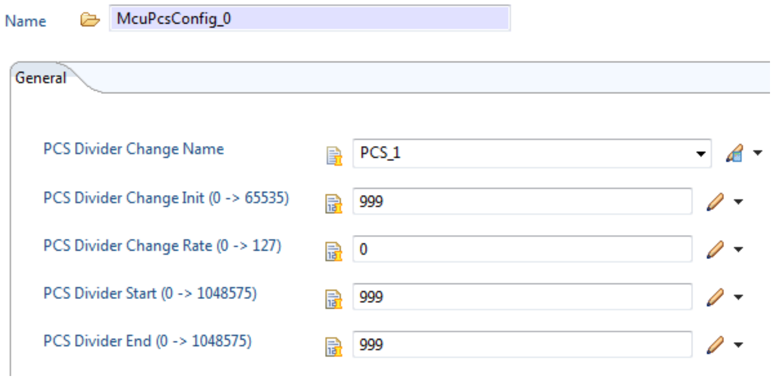
Table 4-283. Attribute McuProgresSwitchDuration (McuProgClkSwitch) detailed description

Property	Value
Label	Progressive Switch Duration
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.35.2 Form McuPcsConfig

This register defines the rate of frequency change and initial change value for the progressive system clock switching when switching the system clock source to or from the XOSC_CLK on ramp-up and ramp-down, respectively Note: Implementation Specific Container.

Is included by form : Form McuProgClkSwitch



Name: McuPcsConfig_0

General

PCS Divider Change Name: PCS_1

PCS Divider Change Init (0 -> 65535): 999

PCS Divider Change Rate (0 -> 127): 0

PCS Divider Start (0 -> 1048575): 999

PCS Divider End (0 -> 1048575): 999

Figure 4-43. Tresos Plugin snapshot for McuPcsConfig form.

4.9.18.35.2.1 McuPCS_Name (McuPcsConfig)

CGM_PCS_DIVCx register configuration name. This is the name of the PCS module. Note: Implementation Specific Parameter.

Table 4-284. Attribute McuPCS_Name (McuPcsConfig) detailed description

Property	Value
Label	PCS Divider Change Name
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.35.2.2 McuPCS_DivInit (McuPcsConfig)

CGM_PCS_DIVCx[INIT] register configuration. This is initial change value of the clock divider for the clock ramp-up phase when switching to the 8-40 MHz crystal osc. Note: Implementation Specific Parameter.

Table 4-285. Attribute McuPCS_DivInit (McuPcsConfig) detailed description

Property	Value
Label	PCS Divider Change Init
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	999

4.9.18.35.2.3 McuPCS_DivRate (McuPcsConfig)

CGM_PCS_DIVCx[RATE] register configuration. This value controls the change value of the clock divider for the clock ramp-up and ramp-down phase when switching to/from the 8-40 MHz crystal osc. Note: Implementation Specific Parameter.

Table 4-286. Attribute McuPCS_DivRate (McuPcsConfig) detailed description

Property	Value
Label	PCS Divider Change Rate
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.35.2.4 McuPCS_DivStart (McuPcsConfig)

CGM_PCS_DIVSx register configuration. This is the start value of the clock divider for the clock ramp-up phase when switching to the crystal osc. Note: Implementation Specific Parameter.

Table 4-287. Attribute McuPCS_DivStart (McuPcsConfig) detailed description

Property	Value
Label	PCS Divider Start
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	999

4.9.18.35.2.5 McuPCS_DivEnd (McuPcsConfig)

CGM_PCS_DIVSx register configuration. Divider End Value - This is the clock divider end value for the clock ramp-down phase when switching from the crystal osc. Note: Implementation Specific Parameter.

Table 4-288. Attribute McuPCS_DivEnd (McuPcsConfig) detailed description

Property	Value
Label	PCS Divider End
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	999

4.9.18.36 Form McuCLKOUT1

This container enables and selects the configuration clocks for CLKOUT1.

Note: Implementation Specific Parameter.

Note: The configuration for CLKOUT1 is implemented following workaround of errata ERR010609 (CLKOUT_1 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational)

Steps to configure for CLKOUT1 is implemented the following:

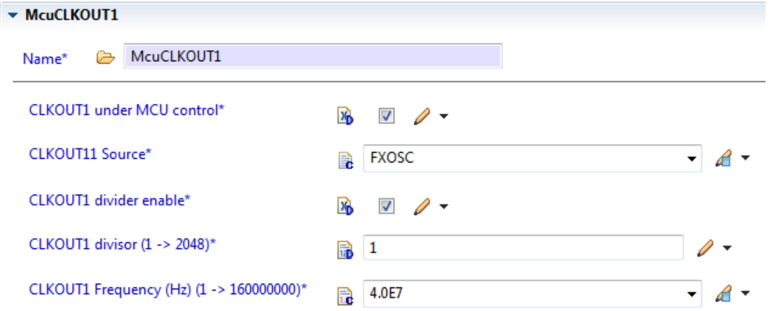
1. Disable the CLKOUT_1 clock divider by writing to MC_CGM_CLKOUT1_DC0[DE] = 0b0

2. Change the CLKOUT_1 clock source selection to FIRC (MC_CGM_CLKOUT1_SC[SELCTL] = 0b1001)

3. Select the desired clock source as the CLKOUT_1

4. Configure and enable the corresponding CLKOUT_1 clock divider by writing to MC_CGM_CLKOUT1_DC0[DE] = 0b1

Is included by form : Form McuClockSettingConfig



McuCLKOUT1

Name*	McuCLKOUT1
CLKOUT1 under MCU control*	☑
CLKOUT11 Source*	FXOSC
CLKOUT1 divider enable*	☑
CLKOUT1 divisor (1 -> 2048)*	1
CLKOUT1 Frequency (Hz) (1 -> 160000000)*	4.0E7

Figure 4-44. Tresos Plugin snapshot for McuCLKOUT1 form.

4.9.18.36.1 McuCLKOUT1UnderMcuControl (McuCLKOUT1)

Set this to TRUE if CLKOUT1 is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-289. Attribute McuCLKOUT1UnderMcuControl (McuCLKOUT1) detailed description

Property	Value
Label	CLKOUT1 under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.36.2 McuCLKOUT1_Source (McuCLKOUT1)

Select the CLKOUT1 Source Selection.

Configure the CGM_CLKOUT1_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-290. Attribute McuCLKOUT1_Source (McuCLKOUT1) detailed description

Property	Value
Label	CLKOUT11 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.36.3 McuCLKOUT1Div_En (McuCLKOUT1)

CGM_CLKOUT1_DC[DE] field register.

Enable the CLKOUT1 divider.

Note: Implementation Specific Parameter.

Table 4-291. Attribute McuCLKOUT1Div_En (McuCLKOUT1) detailed description

Property	Value
Label	CLKOUT1 divider enable
Type	BOOLEAN
Origin	Custom

Table continues on the next page...

Table 4-291. Attribute McuCLKOUT1Div_En (McuCLKOUT1) detailed description (continued)

Property	Value
Symbolic Name	false
Default	true

4.9.18.36.4 McuCLKOUT1_Divisor (McuCLKOUT1)

Set CLKOUT1 Division value.

Set the CGM_CLKOUT1_DC[DIV] field register with value (McuCLKOUT1_Divisor - 1).

This parameter is enabled only if McuCLKOUT1Div_En is true.

Note: Implementation Specific Parameter.

Table 4-292. Attribute McuCLKOUT1_Divisor (McuCLKOUT1) detailed description

Property	Value
Label	CLKOUT1 divisor
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.36.5 McuCLKOUT1Freq (McuCLKOUT1)

This is the frequency after CLKOUT1. Value calculated for user info. It is given in Hz.

Note1: For the clock refs: RTC, LPU, FXOSC_ANALOG, CAN0_PE the frequency must be manually edited. These clocks are not under MCU control. The user is responsible for determining the correct value.

Note2: Implementation Specific Parameter.

Table 4-293. Attribute McuCLKOUT1Freq (McuCLKOUT1) detailed description

Property	Value
Label	CLKOUT1 Frequency (Hz)

Table continues on the next page...

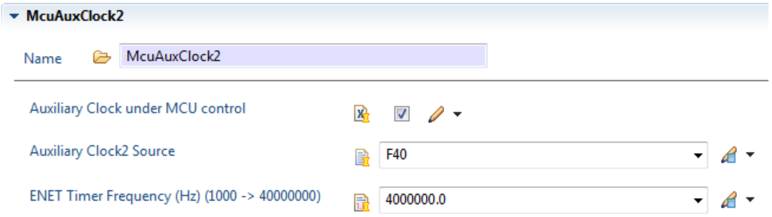
Table 4-293. Attribute McuCLKOUT1Freq (McuCLKOUT1) detailed description (continued)

Property	Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.37 Form McuAuxClock2

This container enables and selects the configuration clocks for ENET timer. Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuAuxClock2

Name McuAuxClock2

Auxiliary Clock under MCU control ☑

Auxiliary Clock2 Source F40

ENET Timer Frequency (Hz) (1000 -> 40000000) 4000000.0

Figure 4-45. Tresos Plugin snapshot for McuAuxClock2 form.

4.9.18.37.1 McuAuxClockUnderMcuControl (McuAuxClock2)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-294. Attribute McuAuxClockUnderMcuControl (McuAuxClock2) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.37.2 McuAuxClk_Source (McuAuxClock2)

Select the Auxiliary Clock 2 Source Selection.

Configure the CGM_AC2_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-295. Attribute McuAuxClk_Source (McuAuxClock2) detailed description

Property	Value
Label	Auxiliary Clock2 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.37.3 McuAuxClkFreq (McuAuxClock2)

This is the frequency after AC2 (ENET timer reference clock). Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

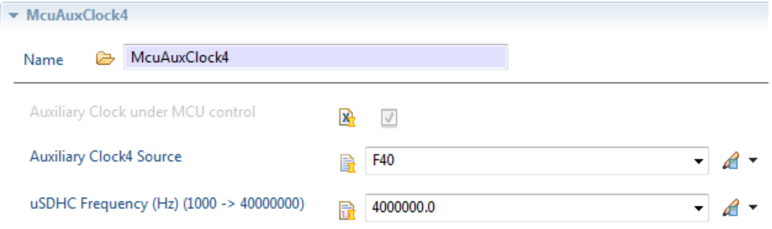
Table 4-296. Attribute McuAuxClkFreq (McuAuxClock2) detailed description

Property	Value
Label	ENET Timer Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.38 Form McuAuxClock4

This container enables and selects the configuration clocks for uSDHC. Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuAuxClock4

Name: McuAuxClock4

Auxiliary Clock under MCU control: ☑

Auxiliary Clock4 Source: F40

uSDHC Frequency (Hz) (1000 -> 40000000): 4000000.0

Figure 4-46. Tresos Plugin snapshot for McuAuxClock4 form.

4.9.18.38.1 McuAuxClockUnderMcuControl (McuAuxClock4)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-297. Attribute McuAuxClockUnderMcuControl (McuAuxClock4) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false

4.9.18.38.2 McuAuxClk_Source (McuAuxClock4)

Select the Auxiliary Clock 4 Source Selection. Configure the CGM_AC4_SC[SELCTL] field register. Note: Implementation Specific Parameter.

Table 4-298. Attribute McuAuxClk_Source (McuAuxClock4) detailed description

Property	Value
Label	Auxiliary Clock4 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.38.3 McuAuxClkFreq (McuAuxClock4)

This is the frequency after AC4 (uSDHC reference clock). Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

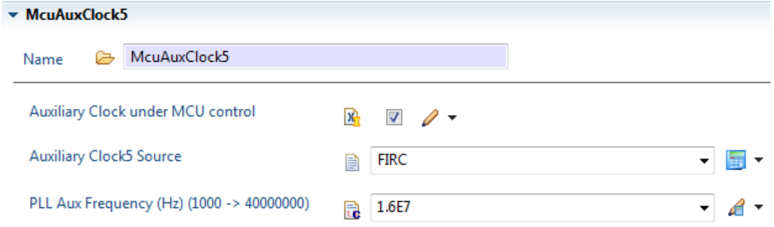
Table 4-299. Attribute McuAuxClkFreq (McuAuxClock4) detailed description

Property	Value
Label	uSDHC Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.39 Form McuAuxClock5

This container enables and selects the configuration clocks for PLL Aux clock. Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuAuxClock5

Name McuAuxClock5

Auxiliary Clock under MCU control ☑

Auxiliary Clock5 Source FIRC

PLL Aux Frequency (Hz) (1000 -> 40000000) 1.6E7

Figure 4-47. Tresos Plugin snapshot for McuAuxClock5 form.

4.9.18.39.1 McuAuxClockUnderMcuControl (McuAuxClock5)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-300. Attribute McuAuxClockUnderMcuControl (McuAuxClock5) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.39.2 McuAuxClk_Source (McuAuxClock5)

Select the Auxiliary Clock 2 Source Selection.

Configure the CGM_AC2_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-301. Attribute McuAuxClk_Source (McuAuxClock5) detailed description

Property	Value
Label	Auxiliary Clock5 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.39.3 McuAuxClkFreq (McuAuxClock5)

This is the frequency after AC5 (PLL Aux clock). Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-302. Attribute McuAuxClkFreq (McuAuxClock5) detailed description

Property	Value
Label	PLL Aux Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.40 Form McuAuxClock6

This container enables and selects the configuration clocks for CLKOUT_0.

Note: Implementation Specific Parameter.

Note: The configuration for CLKOUT0 is implemented following workaround of errata ERR010609 (CLKOUT_0 dividers may become stuck if clock selection is changed while dividers with divide by 2 are operational)

Steps to configure for CLKOUT0 is implemented the following:

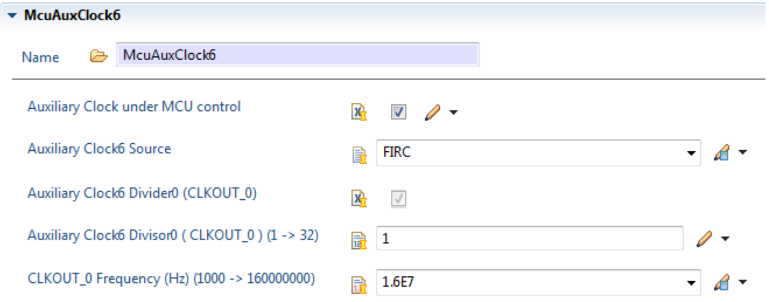
1. Disable the CLKOUT_0 clock divider by writing to MC_CGM_AC6_DC0[DE] = 0b0

2. Change the CLKOUT_0 clock source selection to FIRC (MC_CGM_AC6_SC[SELCTL] = 0b0001)

3. Select the desired clock source as the CLKOUT_0

4. Configure and enable the corresponding CLKOUT_0 clock divider by writing to MC_CGM_AC6_DC0[DE] = 0b1

Is included by form : Form McuClockSettingConfig



McuAuxClock6

Name: McuAuxClock6

Auxiliary Clock under MCU control ☑

Auxiliary Clock6 Source: FIRC

Auxiliary Clock6 Divider0 (CLKOUT_0) ☑

Auxiliary Clock6 Divisor0 (CLKOUT_0) (1 -> 32): 1

CLKOUT_0 Frequency (Hz) (1000 -> 160000000): 1.6E7

Figure 4-48. Tresos Plugin snapshot for McuAuxClock6 form.

4.9.18.40.1 McuAuxClockUnderMcuControl (McuAuxClock6)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-303. Attribute McuAuxClockUnderMcuControl (McuAuxClock6) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.40.2 McuAuxClk_Source (McuAuxClock6)

Select the Auxiliary Clock 2 Source Selection.

Configure the CGM_AC2_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-304. Attribute McuAuxClk_Source (McuAuxClock6) detailed description

Property	Value
Label	Auxiliary Clock6 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.40.3 McuAuxClkDiv0_En (McuAuxClock6)

CGM_AC6_DC0[DE] field register.

Enable the Aux Clock for CLKOUT_0.

Note: Implementation Specific Parameter.

Table 4-305. Attribute McuAuxClkDiv0_En (McuAuxClock6) detailed description

Property	Value
Label	Auxiliary Clock6 Divider0 (CLKOUT_0)
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.40.4 McuAuxClkDiv0_Divisor (McuAuxClock6)

Set Aux6 Divider 0 Division value.

Set the CGM_AC6_DC0[DIV] field register with value (McuAuxClkDiv0_Divisor - 1).

This parameter is enabled only if McuAuxClkDiv0_En is true.

Note: Implementation Specific Parameter.

Table 4-306. Attribute McuAuxClkDiv0_Divisor (McuAuxClock6) detailed description

Property	Value
Label	Auxiliary Clock6 Divisor0 (CLKOUT_0)
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

4.9.18.40.5 McuAuxClkDiv0Freq (McuAuxClock6)

This is the frequency after AuxClk6Div0 (CLKOUT_0 clock). Value calculated for user info. It is given in Hz.

Note1: The RTC clock is not under MCU control. The user is responsible for the correct value.

Note2: Implementation Specific Parameter.

Table 4-307. Attribute McuAuxClkDiv0Freq (McuAuxClock6) detailed description

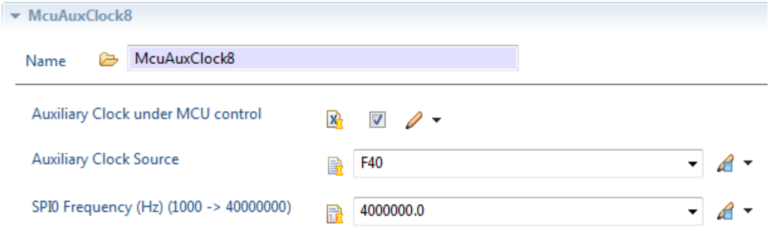
Property	Value
Label	CLKOUT_0 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.41 Form McuAuxClock8

This container enables and selects the configuration clocks for SPI0.

Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuAuxClock8

Name: McuAuxClock8

Auxiliary Clock under MCU control: ☑

Auxiliary Clock Source: F40

SPI0 Frequency (Hz) (1000 -> 40000000): 4000000.0

Figure 4-49. Tresos Plugin snapshot for McuAuxClock8 form.

4.9.18.41.1 McuAuxClockUnderMcuControl (McuAuxClock8)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-308. Attribute McuAuxClockUnderMcuControl (McuAuxClock8) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.41.2 McuAuxClk_Source (McuAuxClock8)

Select the Auxiliary Clock 8 Source Selection.

Configure the CGM_AC8_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-309. Attribute McuAuxClk_Source (McuAuxClock8) detailed description

Property	Value
Label	Auxiliary Clock Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.41.3 McuAuxClkFreq (McuAuxClock8)

This is the frequency after AC8 (SPI0 reference clock). Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-310. Attribute McuAuxClkFreq (McuAuxClock8) detailed description

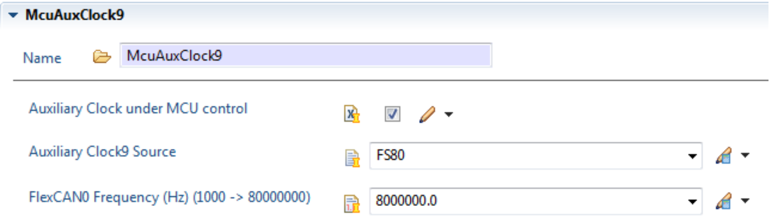
Property	Value
Label	SPI0 Frequency (Hz)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.42 Form McuAuxClock9

This container enables and selects the configuration clocks for FlexCAN0.

Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuAuxClock9

Name: McuAuxClock9

Auxiliary Clock under MCU control ☑

Auxiliary Clock9 Source: FS80

FlexCAN0 Frequency (Hz) (1000 -> 80000000): 8000000.0

Figure 4-50. Tresos Plugin snapshot for McuAuxClock9 form.

4.9.18.42.1 McuAuxClockUnderMcuControl (McuAuxClock9)

Set this to TRUE if this aux clock is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-311. Attribute McuAuxClockUnderMcuControl (McuAuxClock9) detailed description

Property	Value
Label	Auxiliary Clock under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.42.2 McuAuxClk_Source (McuAuxClock9)

Select the Auxiliary Clock 9 Source Selection.

Configure the CGM_AC9_SC[SELCTL] field register.

Note: Implementation Specific Parameter.

Table 4-312. Attribute McuAuxClk_Source (McuAuxClock9) detailed description

Property	Value
Label	Auxiliary Clock9 Source
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.18.42.3 McuAuxClkFreq (McuAuxClock9)

This is the frequency after AC9 (FlexCAN0 reference clock). Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-313. Attribute McuAuxClkFreq (McuAuxClock9) detailed description

Property	Value
Label	FlexCAN0 Frequency (Hz)

Table continues on the next page...

Table 4-313. Attribute McuAuxClkFreq (McuAuxClock9) detailed description (continued)

Property	Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

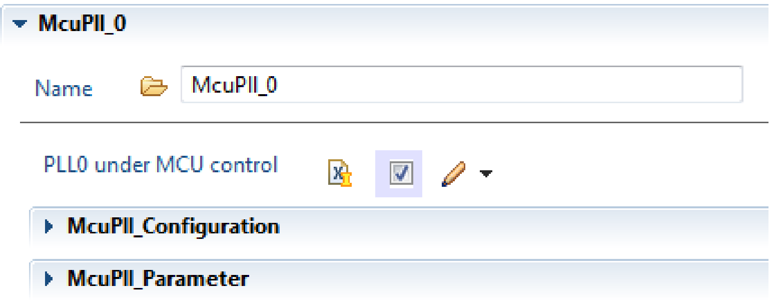
4.9.18.43 Form McuPll_0

This container provides the specific configuration for the PLL0. The PLL can be switched off when not required to achieve lower consumption by programming the ME_x_MC registers in the MC_ME module. Note: Implementation Specific Container.

Is included by form : Form McuClockSettingConfig

Included forms :

- Form McuPll_Configuration
- Form McuPll_Parameter



McuPll_0

Name McuPll_0

PLL0 under MCU control ☑

McuPll_Configuration

McuPll_Parameter

Figure 4-51. Tresos Plugin snapshot for McuPll_0 form.

4.9.18.43.1 McuPLLUnderMcuControl (McuPll_0)

Set this to TRUE if this PLL is under mcu control

If it is FALSE then the mcu driver will not write the corresponding registers.

The user must still set the values - they are used by the clock refference points

Note: Implementation Specific Parameter.

Table 4-314. Attribute McuPLLUnderMcuControl (McuPll_0) detailed description

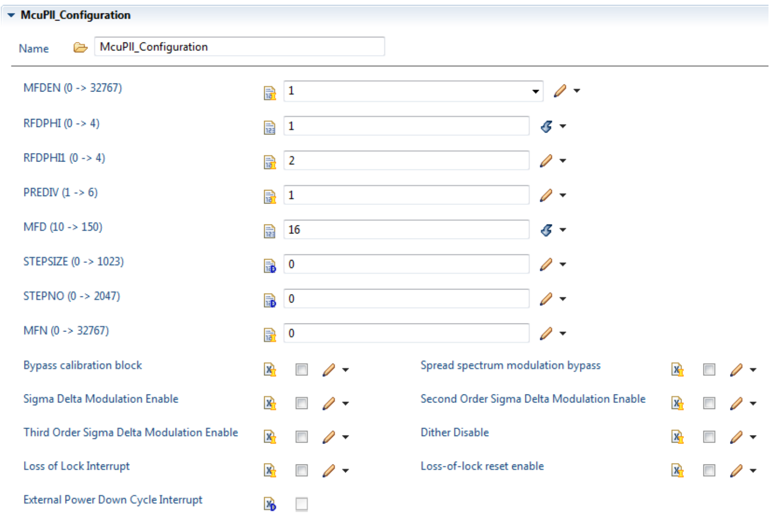
Property	Value
Label	PLL0 under MCU control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.43.2 Form McuPll_Configuration

Configuration values for PLL.

Note: Implementation Specific Parameter.

Is included by form : Form McuPll_0



McuPll_Configuration

Name	McuPll_Configuration
MFDEN (0 -> 32767)	1
RFDPHI (0 -> 4)	1
RFDPHI1 (0 -> 4)	2
PREDIV (1 -> 6)	1
MFD (10 -> 150)	16
STEPSIZE (0 -> 1023)	0
STEPNO (0 -> 2047)	0
MFN (0 -> 32767)	0

Bypass calibration block	□	Spread spectrum modulation bypass	□
Sigma Delta Modulation Enable	□	Second Order Sigma Delta Modulation Enable	□
Third Order Sigma Delta Modulation Enable	□	Dither Disable	□
Loss of Lock Interrupt	□	Loss-of-lock reset enable	□
External Power Down Cycle Interrupt	□		

Figure 4-52. Tresos Plugin snapshot for McuPll_Configuration form.

4.9.18.43.2.1 McuPllCal3Mfden (McuPll_Configuration)

Denominator of fractional loop division factor - PLLDIG_PLLCAL3[MFDEN].

Note: The value for this field must be greater than PLLFD[MFN]

. Note: Implementation Specific Parameter.

Table 4-315. Attribute McuPllCal3Mfden (McuPll_Configuration) detailed description

Property	Value
Label	MFDEN
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4096
Invalid	Range <=32767 >=0

4.9.18.43.2.2 McuPllDvRfdphi (McuPll_Configuration)

Set PLL PHI reduced frequency divider.

Sets the PLL: PLLDIG_PLLDV[RFDPHI] field register.

Valid value range is:

Table 4-316. Detailed description.

Value	Meaning
0	Divide by 2
1	Divide by 4
2	Divide by 8
3	Divide by 16
4	Divide by 32

Note: Implementation Specific Parameter.

Table 4-317. Attribute McuPllDvRfdphi (McuPll_Configuration) detailed description

Property	Value
Label	RFDPHI
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range <=4 >=0

4.9.18.43.2.3 McuPllDvRfdphi1 (McuPll_Configuration)

Set PLL PHI1 reduced frequency divider.

Sets the PLL: PLLDIG_PLLDV[RFDPHI1] field register.

Valid value range is:

Table 4-318. Detailed description.

Value	Meaning
0	Divide by 2
1	Divide by 4
2	Divide by 8
3	Divide by 16
4	Divide by 32

Note: Implementation Specific Parameter.

Table 4-319. Attribute McuPllDvRfdphi1 (McuPll_Configuration) detailed description

Property	Value
Label	RFDPHI1
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4
Invalid	Range <=4 >=0

4.9.18.43.2.4 McuPllDvPrediv (McuPll_Configuration)

Input clock predivider.

Sets the PLL: PLLDIG_PLLDV[PREDIV] field register.

This field controls the value of the divider on the input clock. The output of the predivider circuit generates the reference clock to the PLL analog loop.

Note: Implementation Specific Parameter.

Table 4-320. Attribute McuPllDvPrediv (McuPll_Configuration) detailed description

Property	Value
Label	PREDIV
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range <=6 >=1

4.9.18.43.2.5 McuPllDvMfd (McuPll_Configuration)

Loop multiplication factor divider.

Sets the PLL: PLLDIG_PLLDV[MFD] field register.

This field controls the value of the divider in the feedback loop.

The value specified by the MFD bits establishes the multiplication factor applied to the reference frequency.

Divider value = MFD, where MFD has the range 10...150 (Ah...96h). All other values are reserved.

Note: Implementation Specific Parameter.

Table 4-321. Attribute McuPllDvMfd (McuPll_Configuration) detailed description

Property	Value
Label	MFD
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	18
Invalid	Range <=150 >=10

4.9.18.43.2.6 McuPllFmStepsize (McuPll_Configuration)

Modulation period.

Sets the PLL: PLLDIG_PLLFM[STEPSIZE] field register.

STEPSIZE is the binary equivalent of the modulation period variable

Note: Implementation Specific Parameter.

Table 4-322. Attribute McuPllFmStepsize (McuPll_Configuration) detailed description

Property	Value
Label	STEPSIZE
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=1023 >=0

4.9.18.43.2.7 McuPllFmStepno (McuPll_Configuration)

Increment step.

Sets the PLL: PLLDIG_PLLFM[STEPNO] field register.

This field is the binary equivalent of the STEPNO variable.

Note: Implementation Specific Parameter.

Table 4-323. Attribute McuPllFmStepno (McuPll_Configuration) detailed description

Property	Value
Label	STEPNO
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=2047 >=0

4.9.18.43.2.8 McuPllFdMfn (McuPll_Configuration)

Numerator for fractional loop division factor - PLLDIG_PLLFD[MFN].

Note: Implementation Specific Parameter.

Table 4-324. Attribute McuPllFdMfn (McuPll_Configuration) detailed description

Property	Value
Label	MFN
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0
Invalid	Range <=32767 >=0

4.9.18.43.2.9 McuPllCalBypcal (McuPll_Configuration)

Bypass calibration block.

Set the PLL: PLLDIG_PLLCAL1[BYPCAL] field.

0 - Do not bypass the analog PLL calibration block.

1 - Bypass the analog PLL calibration block.

Note: Implementation Specific Parameter.

Table 4-325. Attribute McuPllCalBypcal (McuPll_Configuration) detailed description

Property	Value
Label	Bypass calibration block
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.10 McuPllFmSscgbyp (McuPll_Configuration)

Modulation enable.

This bit enables spectrum modulation.

Set the PLL: PLLDIG_PLLFM[SSCGBYP] field.

0 - Spread spectrum modulation is not bypassed.

1 - Spread spectrum modulation is bypassed.

Note: Implementation Specific Parameter.

Table 4-326. Attribute McuPllFmSscgbyp (McuPll_Configuration) detailed description

Property	Value
Label	Spread spectrum modulation bypass
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.43.2.11 McuPllFdSmden (McuPll_Configuration)

Sigma Delta Modulation Enable.

Set the PLL: PLLDIG_PLLFD[SMDEN] field register.

0 - Sigma delta modulation enabled.

1 - Sigma delta modulation disabled.

Note: Implementation Specific Parameter.

Table 4-327. Attribute McuPllFdSmden (McuPll_Configuration) detailed description

Property	Value
Label	Sigma Delta Modulation Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.12 McuPllFdSdm2 (McuPll_Configuration)

Second Order Sigma Delta Modulation Select.

Set the PLL: PLLDIG_PLLFD[SDM2] field register.

0 - 2nd order sigma delta modulation disabled.

1 - 2nd order sigma delta modulation enabled.

Note: Implementation Specific Parameter.

Table 4-328. Attribute McuPllFdSdm2 (McuPll_Configuration) detailed description

Property	Value
Label	Second Order Sigma Delta Modulation Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.13 McuPllFdSdm3 (McuPll_Configuration)

Third Order Sigma Delta Modulation Select.

Set the PLL: PLLDIG_PLLFD[SDM3] field register.

0 - 3nd order sigma delta modulation disabled.

1 - 3nd order sigma delta modulation enabled.

Note: Implementation Specific Parameter.

Table 4-329. Attribute McuPllFdSdm3 (McuPll_Configuration) detailed description

Property	Value
Label	Third Order Sigma Delta Modulation Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.14 McuPllFdDthDis (McuPll_Configuration)

Dither Disable.

Set the PLL: PLLDIG_PLLFD[DTHDIS] field register.

0 - Dither enabled.

1 - Dither disabled.

Note: Implementation Specific Parameter.

Table 4-330. Attribute McuPllFdDthDis (McuPll_Configuration) detailed description

Property	Value
Label	Dither Disable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.15 McuPllCrLolie (McuPll_Configuration)

Loss of lock interrupt enable. Set the PLL: PLLDIG_PLLCR[LOLIE] field register. 0 - Interrupt disabled. 1 - Interrupt enabled. Note: Implementation Specific Parameter.

Table 4-331. Attribute McuPllCrLolie (McuPll_Configuration) detailed description

Property	Value
Label	Loss of Lock Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.16 McuPllCrLolre (McuPll_Configuration)

Loss-of-lock reset enable. Set the PLL: PLLDIG_PLLCR[LOLRE] field register. 0 - Ignore loss-of-lock. Reset not asserted.

1 - Assert reset on loss-of-lock when operating in normal mode. Note: Implementation Specific Parameter.

Table 4-332. Attribute McuPllCrLolre (McuPll_Configuration) detailed description

Property	Value
Label	Loss-of-lock reset enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.2.17 McuExternalPowerDownCycleInt (McuPll_Configuration)

External Power Down cycle Complete indication interrupt enable.

Configure the PLLDIG_PLLCR[EXPDIE] bit.

0 - Interrupt disabled.

1 - Interrupt enabled.

Note: Implementation Specific Parameter.

Table 4-333. Attribute McuExternalPowerDownCycleInt (McuPll_Configuration) detailed description

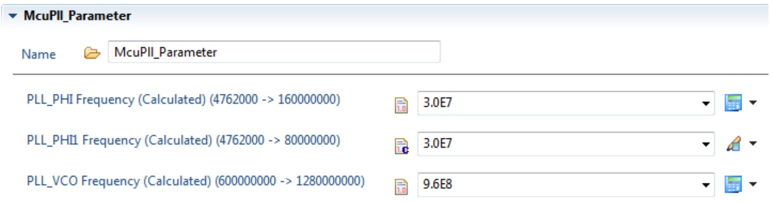
Property	Value
Label	External Power Down Cycle Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.43.3 Form McuPll_Parameter

Calculated values for PLL.

Note: Implementation Specific Parameter.

Is included by form : Form McuPll_0



McuPll_Parameter

Name	McuPll_Parameter
PLL_PHI Frequency (Calculated) (4762000 -> 160000000)	3.0E7
PLL_PHI1 Frequency (Calculated) (4762000 -> 80000000)	3.0E7
PLL_VCO Frequency (Calculated) (600000000 -> 1280000000)	9.6E8

Figure 4-53. Tresos Plugin snapshot for McuPll_Parameter form.

4.9.18.43.3.1 PLL_PHI_Frequency (McuPll_Parameter)

Output value for

PLL_PHI frequency

.

The valid range is [4.762 ... 160] MHz.

Note: Implementation Specific Parameter.

Table 4-334. Attribute PLL_PHI_Frequency (McuPll_Parameter) detailed description

Property	Value
Label	PLL_PHI Frequency (Calculated)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.43.3.2 PLL_PHI1_Frequency (McuPll_Parameter)

Output value for

PLL_PHI1 frequency

.

The valid range is [4.762 ... 80] MHz.

Note: Implementation Specific Parameter.

Table 4-335. Attribute PLL_PHI1_Frequency (McuPll_Parameter) detailed description

Property	Value
Label	PLL_PHI1 Frequency (Calculated)

Table continues on the next page...

Table 4-335. Attribute PLL_PHI1_Frequency (McuPll_Parameter) detailed description (continued)

Property	Value
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.43.3.3 PLL_VCO_Frequency (McuPll_Parameter)

Output value for

PLL_VCO frequency

.

The valid range is [600 ... 1280] MHz.

Note: Implementation Specific Parameter.

Table 4-336. Attribute PLL_VCO_Frequency (McuPll_Parameter) detailed description

Property	Value
Label	PLL_VCO Frequency (Calculated)
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.44 Form McuEMIOS

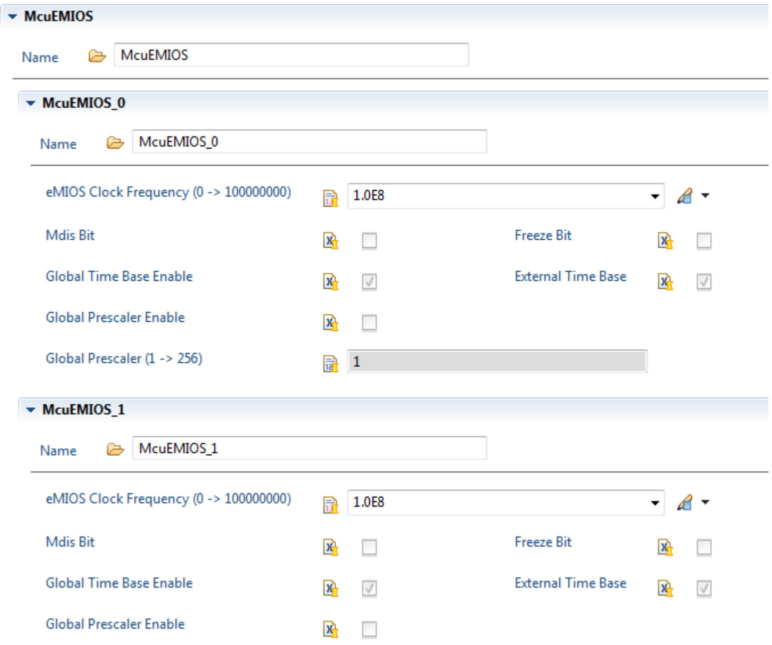
This container controls eMIOS.

Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig

Included forms :

- Form McuEMIOS_0
- Form McuEMIOS_1
- Form McuEMIOS_2



McuEMIOS

Name: McuEMIOS

McuEMIOS_0

Name: McuEMIOS_0

eMIOS Clock Frequency (0 -> 100000000): 1.0E8

Mdis Bit □ Freeze Bit □

Global Time Base Enable ☑ External Time Base ☑

Global Prescaler Enable □

Global Prescaler (1 -> 256): 1

McuEMIOS_1

Name: McuEMIOS_1

eMIOS Clock Frequency (0 -> 100000000): 1.0E8

Mdis Bit □ Freeze Bit □

Global Time Base Enable ☑ External Time Base ☑

Global Prescaler Enable □

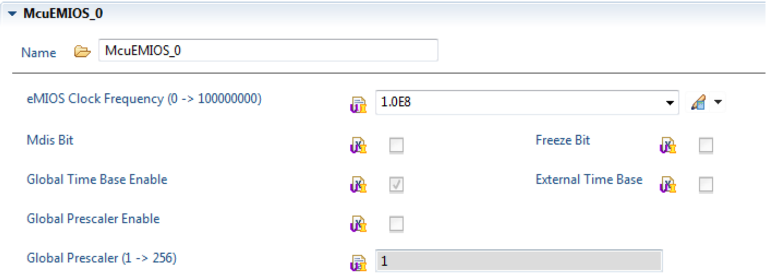
Figure 4-54. Tresos Plugin snapshot for McuEMIOS form.

4.9.18.44.1 Form McuEMIOS_0

This container controls eMIOS 0.

Note: Implementation Specific Parameter.

Is included by form : Form McuEMIOS



McuEMIOS_0

Name: McuEMIOS_0

eMIOS Clock Frequency (0 -> 100000000): 1.0E8

Mdis Bit □ Freeze Bit □

Global Time Base Enable ☑ External Time Base □

Global Prescaler Enable □

Global Prescaler (1 -> 256): 1

Figure 4-55. Tresos Plugin snapshot for McuEMIOS_0 form.

4.9.18.44.1.1 McuEMIOS_Clk (McuEMIOS_0)

This is the frequency for the specific instance of the "McuClockReferencePoint" container. Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-337. Attribute McuEMIOS_Clk (McuEMIOS_0) detailed description

Property	Value
Label	eMIOS Clock Frequency
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.44.1.2 McuMdisBit (McuEMIOS_0)

Check this to put the eMIOS200 in low power mode. The MDIS bit is used to stop the clock of the block, except the access to registers EMIOSMCR, EMIOSOUDIS and EMIOSUCDIS.

Note:implementation specific parameter.

Table 4-338. Attribute McuMdisBit (McuEMIOS_0) detailed description

Property	Value
Label	Mdis Bit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.1.3 McuFreezeBit (McuEMIOS_0)

Freeze Bit (FRZ in EMIOS_MCR_A register).Enable the eMIOS200 to freeze the registers of the unified channels when debug mode is requested at MCU level. Each unified channel must have FREN bit set in order to enter freeze mode.

Table 4-339. Attribute McuFreezeBit (McuEMIOS_0) detailed description

Property	Value
Label	Freeze Bit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.1.4 McuGlobalTimeBaseEnable (McuEMIOS_0)

Global Time Base Enable Bit (GTBE in EMIOS_MCR register).The GTBE bit is used to export a global time base enable from the module and provide a method to start time bases of several blocks simultaneously.

Table 4-340. Attribute McuGlobalTimeBaseEnable (McuEMIOS_0) detailed description

Property	Value
Label	Global Time Base Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.44.1.5 McuExternalTimeBase (McuEMIOS_0)

The ETB bit selects the time base source that drives counter bus.

Table 4-341. Attribute McuExternalTimeBase (McuEMIOS_0) detailed description

Property	Value
Label	External Time Base
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.1.6 McuGlobalPrescalerEnable (McuEMIOS_0)

Select the clock source clock.

Note:implementation specific parameter.

Table 4-342. Attribute McuGlobalPrescalerEnable (McuEMIOS_0) detailed description

Property	Value
Label	Global Prescaler Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.1.7 McuGlobalPrescaler (McuEMIOS_0)

Global Prescaler Bits (GPRE[0:7] in EMIOS_MCR register).

The bits select the clock divider value for the global prescaler.

The allowed values are 0 up to 255 (divide ratio 1 up to 256).

Note:implementation specific parameter.

Table 4-343. Attribute McuGlobalPrescaler (McuEMIOS_0) detailed description

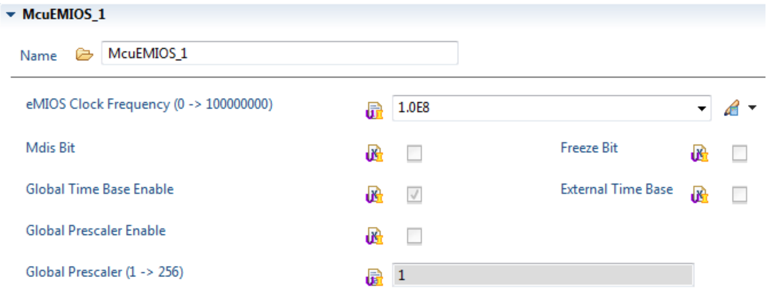
Property	Value
Label	Global Prescaler
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range <=256 >=1

4.9.18.44.2 Form McuEMIOS_1

This container controls eMIOS 1.

Note: Implementation Specific Parameter.

Is included by form : Form McuEMIOS



McuEMIOS_1

Name McuEMIOS_1

eMIOS Clock Frequency (0 -> 100000000) 1.0E8

Mdis Bit □ Freeze Bit □

Global Time Base Enable ☑ External Time Base □

Global Prescaler Enable □

Global Prescaler (1 -> 256) 1

Figure 4-56. Tresos Plugin snapshot for McuEMIOS_1 form.

4.9.18.44.2.1 McuEMIOS_Clk (McuEMIOS_1)

This is the frequency for the specific instance of the "McuClockReferencePoint" container. Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-344. Attribute McuEMIOS_Clk (McuEMIOS_1) detailed description

Property	Value
Label	eMIOS Clock Frequency
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.44.2.2 McuMdisBit (McuEMIOS_1)

Check this to put the eMIOS200 in low power mode. The MDIS bit is used to stop the clock of the block, except the access to registers EMIOSMCR, EMIOSOUDIS and EMIOSUCDIS.

Note:implementation specific parameter.

Table 4-345. Attribute McuMdisBit (McuEMIOS_1) detailed description

Property	Value
Label	Mdis Bit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.2.3 McuFreezeBit (McuEMIOS_1)

Freeze Bit (FRZ in EMIOS_MCR_A register).Enable the eMIOS200 to freeze the registers of the unified channels when debug mode is requested at MCU level. Each unified channel must have FREN bit set in order to enter freeze mode.

Table 4-346. Attribute McuFreezeBit (McuEMIOS_1) detailed description

Property	Value
Label	Freeze Bit
Type	BOOLEAN

Table continues on the next page...

Table 4-346. Attribute McuFreezeBit (McuEMIOS_1) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.2.4 McuGlobalTimeBaseEnable (McuEMIOS_1)

Global Time Base Enable Bit (GTBE in EMIOS_MCR register).The GTBE bit is used to export a global time base enable from the module and provide a method to start time bases of several blocks simultaneously.

Table 4-347. Attribute McuGlobalTimeBaseEnable (McuEMIOS_1) detailed description

Property	Value
Label	Global Time Base Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.44.2.5 McuExternalTimeBase (McuEMIOS_1)

The ETB bit selects the time base source that drives counter bus.

Table 4-348. Attribute McuExternalTimeBase (McuEMIOS_1) detailed description

Property	Value
Label	External Time Base
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.2.6 McuGlobalPrescalerEnable (McuEMIOS_1)

Select the clock source clock.

Note:implementation specific parameter.

Table 4-349. Attribute McuGlobalPrescalerEnable (McuEMIOS_1) detailed description

Property	Value
Label	Global Prescaler Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.2.7 McuGlobalPrescaler (McuEMIOS_1)

Global Prescaler Bits (GPRE[0:7] in EMIOS_MCR register).

The bits select the clock divider value for the global prescaler.

The allowed values are 0 up to 255 (divide ratio 1 up to 256).

Note:implementation specific parameter.

Table 4-350. Attribute McuGlobalPrescaler (McuEMIOS_1) detailed description

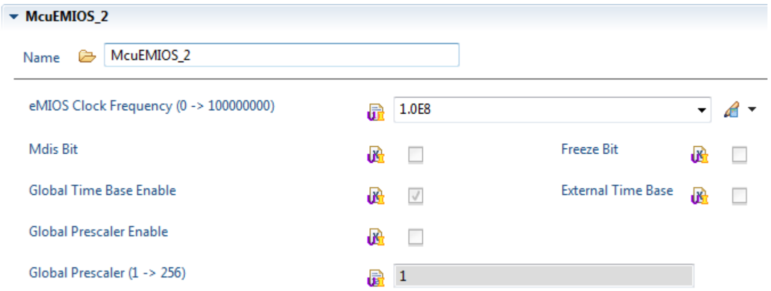
Property	Value
Label	Global Prescaler
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range <=256 >=1

4.9.18.44.3 Form McuEMIOS_2

This container controls eMIOS 2.

Note: Implementation Specific Parameter.

Is included by form : Form McuEMIOS



McuEMIOS_2

Name: McuEMIOS_2

eMIOS Clock Frequency (0 -> 100000000): 1.0E8

Mdis Bit □ Freeze Bit □

Global Time Base Enable ☑ External Time Base □

Global Prescaler Enable □

Global Prescaler (1 -> 256): 1

Figure 4-57. Tresos Plugin snapshot for McuEMIOS_2 form.

4.9.18.44.3.1 McuEMIOS_Clk (McuEMIOS_2)

This is the frequency for the specific instance of the "McuClockReferencePoint" container. Value calculated for user info. It is given in Hz. Note: Implementation Specific Parameter.

Table 4-351. Attribute McuEMIOS_Clk (McuEMIOS_2) detailed description

Property	Value
Label	eMIOS Clock Frequency
Type	FLOAT
Origin	Custom
Symbolic Name	false

4.9.18.44.3.2 McuMdisBit (McuEMIOS_2)

Check this to put the eMIOS200 in low power mode. The MDIS bit is used to stop the clock of the block, except the access to registers EMIOSMCR, EMIOSOUDIS and EMIOSUCDIS.

Note:implementation specific parameter.

Table 4-352. Attribute McuMdisBit (McuEMIOS_2) detailed description

Property	Value
Label	Mdis Bit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.3.3 McuFreezeBit (McuEMIOS_2)

Freeze Bit (FRZ in EMIOS_MCR_A register).Enable the eMIOS200 to freeze the registers of the unified channels when debug mode is requested at MCU level. Each unified channel must have FREN bit set in order to enter freeze mode.

Table 4-353. Attribute McuFreezeBit (McuEMIOS_2) detailed description

Property	Value
Label	Freeze Bit
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.3.4 McuGlobalTimeBaseEnable (McuEMIOS_2)

Global Time Base Enable Bit (GTBE in EMIOS_MCR register).The GTBE bit is used to export a global time base enable from the module and provide a method to start time bases of several blocks simultaneously.

Table 4-354. Attribute McuGlobalTimeBaseEnable (McuEMIOS_2) detailed description

Property	Value
Label	Global Time Base Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.18.44.3.5 McuExternalTimeBase (McuEMIOS_2)

The ETB bit selects the time base source that drives counter bus.

Table 4-355. Attribute McuExternalTimeBase (McuEMIOS_2) detailed description

Property	Value
Label	External Time Base
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.3.6 McuGlobalPrescalerEnable (McuEMIOS_2)

Select the clock source clock.

Note:implementation specific parameter.

Table 4-356. Attribute McuGlobalPrescalerEnable (McuEMIOS_2) detailed description

Property	Value
Label	Global Prescaler Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.44.3.7 McuGlobalPrescaler (McuEMIOS_2)

Global Prescaler Bits (GPRE[0:7] in EMIOS_MCR register).

The bits select the clock divider value for the global prescaler.

The allowed values are 0 up to 255 (divide ratio 1 up to 256).

Note:implementation specific parameter.

Table 4-357. Attribute McuGlobalPrescaler (McuEMIOS_2) detailed description

Property	Value
Label	Global Prescaler
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1
Invalid	Range <=256 >=1

4.9.18.45 Form McuClkMonitor_0

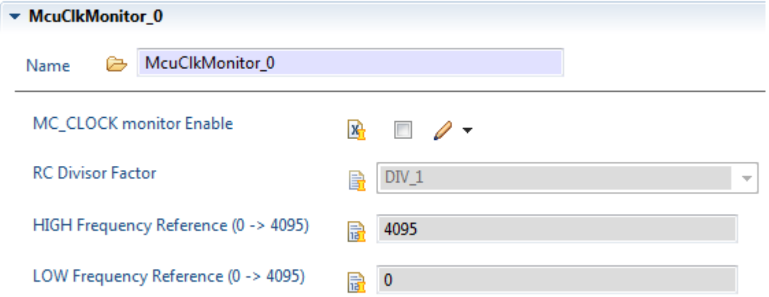
This container contains the specific configuration (parameters) of the Monitor_0.

Clock Monitor Unit for Motor Clock.

This parameter is enabled only if "McuClockSrcFailureNotification" is enabled.

Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuClkMonitor_0

Name: McuClkMonitor_0

MC_CLOCK monitor Enable: □

RC Divisor Factor: DIV_1

HIGH Frequency Reference (0 -> 4095): 4095

LOW Frequency Reference (0 -> 4095): 0

Figure 4-58. Tresos Plugin snapshot for McuClkMonitor_0 form.

4.9.18.45.1 McuClkMonitorEn (McuClkMonitor_0)

Enables/Disables the clock monitor (CMU_CSR[CME]).

Note: Implementation Specific Parameter.

Table 4-358. Attribute McuClkMonitorEn (McuClkMonitor_0) detailed description

Property	Value
Label	MC_CLOCK monitor Enable
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.18.45.2 McuRCDivisorFactor (McuClkMonitor_0)

Set the CMU0: CMU_CSR[RCDIV] field register.

CMU0 detects if the Fxosc is smaller than 16000000/(2^CMU_CSR[RCDIV]) when is used for XOSC clock monitoring.

Note: Implementation Specific Parameter.

Table 4-359. Attribute McuRCDivisorFactor (McuClkMonitor_0) detailed description

Property	Value
Label	RC Divisor Factor
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	DIV_1

4.9.18.45.3 McuHighFrequencyRef (McuClkMonitor_0)

The value of the Max Frequency.

CMU0 detects if the monitored clock is greater than McuHighFrequencyRef.

The reference value is given by: (HFREF /16) * (fCLKMT0_RMN / 4).

Note: Implementation Specific Parameter.

Table 4-360. Attribute McuHighFrequencyRef (McuClkMonitor_0) detailed description

Property	Value
Label	HIGH Frequency Reference
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	4095

4.9.18.45.4 McuLowFrequencyRef (McuClkMonitor_0)

The value of the Min Frequency.

CMU0 detects if the monitored clock is smaller than McuLowFrequencyRef.

The reference value is given by: (LFREF / 16) * (fCLKMT0_RMN / 4).

Note: Implementation Specific Parameter.

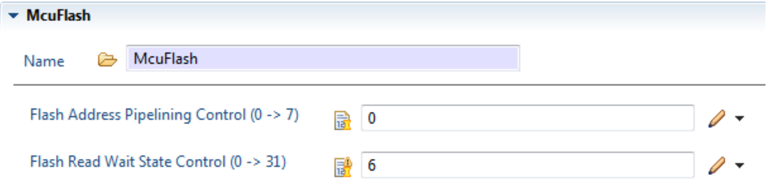
Table 4-361. Attribute McuLowFrequencyRef (McuClkMonitor_0) detailed description

Property	Value
Label	LOW Frequency Reference
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.46 Form McuFlash

This container contains the specific configuration (parameters) of the Mcu Flash PFCR0 register. Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig



McuFlash

Name: McuFlash

Flash Address Pipelining Control (0 -> 7): 0

Flash Read Wait State Control (0 -> 31): 6

Figure 4-59. Tresos Plugin snapshot for McuFlash form.

4.9.18.46.1 McuFlashAddrPipelineControl (McuFlash)

PFCR1[APC] field configuration register. This field controls the number of cycles that a subsequent flash read from the opposite AHB port can be initiated prior to the previous read data being valid.

000 Pipelined access to the flash disabled.

001 A pipelined access can be initiated 1 cycle before the previous data is valid

010 A pipelined access can be initiated 2 cycles before the previous data is valid

011 A pipelined access can be initiated 3 cycles before the previous data is valid

100 Pipelined access to the flash is disabled and one wait state is inserted before a subsequent access can be initiated

101 Pipelined access to the flash is disabled and one wait state is inserted before a subsequent access can be initiated

110 Pipelined access to the flash is disabled and one wait state is inserted before a subsequent access can be initiated

111 Pipelined access to the flash is disabled and one wait state is inserted before a subsequent access can be initiated

Note: Implementation Specific Parameter.

Table 4-362. Attribute McuFlashAddrPipelineControl (McuFlash) detailed description

Property	Value
Label	Flash Address Pipelining Control
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.46.2 McuFlashReadWaitStateControl (McuFlash)

Configure PFCR1[RWSC] field of FLASH module. This field controls the number of wait-states to be added to the best-case flash array access time for reads. The best-case flash array access time for reads is one cycle. 00000 No additional wait-states are added 00001 One additional wait-state is added ... 11111 Thirty-one additional wait-states are added Note: Implementation Specific Parameter.

Table 4-363. Attribute McuFlashReadWaitStateControl (McuFlash) detailed description

Property	Value
Label	Flash Read Wait State Control
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	1

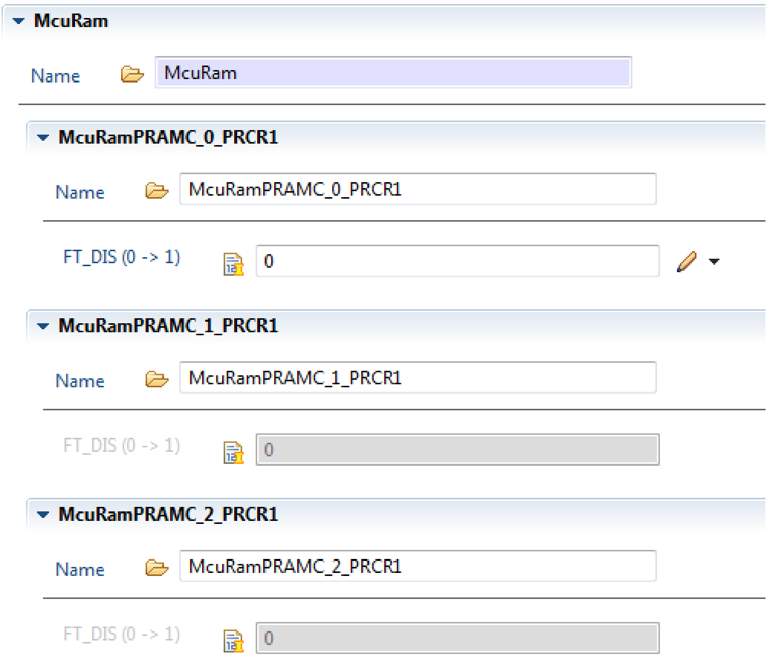
4.9.18.47 Form McuRam

This container contains the specific configuration (parameters) for Platform RAM wait-state control. Note: Implementation Specific Parameter.

Is included by form : Form McuClockSettingConfig

Included forms :

- Form McuRamPRAMC_0_PRCR1
- Form McuRamPRAMC_1_PRCR1
- Form McuRamPRAMC_2_PRCR1



McuRam

Name: McuRam

McuRamPRAMC_0_PRCR1

Name: McuRamPRAMC_0_PRCR1

FT_DIS (0 -> 1): 0

McuRamPRAMC_1_PRCR1

Name: McuRamPRAMC_1_PRCR1

FT_DIS (0 -> 1): 0

McuRamPRAMC_2_PRCR1

Name: McuRamPRAMC_2_PRCR1

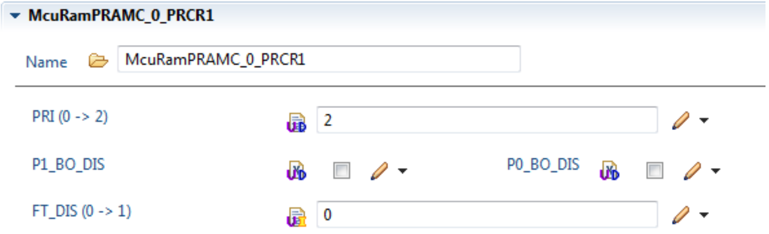
FT_DIS (0 -> 1): 0

Figure 4-60. Tresos Plugin snapshot for McuRam form.

4.9.18.47.1 Form McuRamPRAMC_0_PRCR1

The Platform RAM Configuration Register (PRAMC_0_PRCR1) is used to specify operation of the quad-port RAM controller's internal controller for SRAM bank 0. Note: Implementation Specific Parameter.

Is included by form : Form McuRam



McuRamPRAMC_0_PRCR1

Name: McuRamPRAMC_0_PRCR1

PRI (0 -> 2): 2

P1_BO_DIS ☐ P0_BO_DIS ☐

FT_DIS (0 -> 1): 0

Figure 4-61. Tresos Plugin snapshot for McuRamPRAMC_0_PRCR1 form.

4.9.18.47.1.1 McuFT_DIS (McuRamPRAMC_0_PRCR1)

PRAMC0_PRCR1[FT_DIS] field configuration.

Flow through disabled.

This field defines the AHB response on reads. The state of this field has no impact on the response

latency on writes. This bit is cleared by hardware reset.

NOTE: Do not change the FT_DIS bit value while accessing system RAM. Relocate code programming

the FT_DIS bit to another memory area, e.g., local core memory.

0 RAM read data is passed directly to the system bus, incurring no additional latency

1 RAM read data is registered prior to returning on the system bus, incurring 1 extra cycle of latency

Note: Implementation Specific Parameter.

Table 4-364. Attribute McuPRI (McuRamPRAMC_0_PRCR1) detailed description

Property	Value
Label	PRI
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	2

Table 4-365. Attribute McuP1BODIS (McuRamPRAMC_0_PRCR1) detailed description

Property	Value
Label	P1_BO_DIS
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

Table 4-366. Attribute McuP0BODIS (McuRamPRAMC_0_PRCR1) detailed description

Property	Value
Label	McuP0BODIS
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

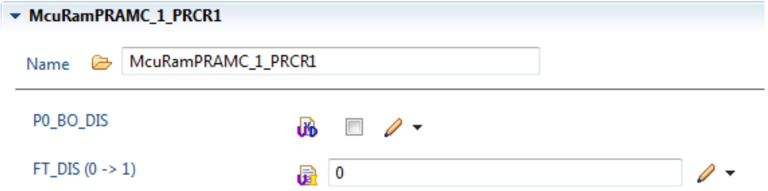
Table 4-367. Attribute McuFT_DIS (McuRamPRAMC_0_PRCR1) detailed description

Property	Value
Label	FT_DIS
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.47.2 Form McuRamPRAMC_1_PRCR1

The Platform RAM Configuration Register (PRAMC_1_PRCR1) is used to specify operation of the quad-port RAM controller's internal controller for SRAM bank 0. Note: Implementation Specific Parameter.

Is included by form : Form McuRam



McuRamPRAMC_1_PRCR1

Name McuRamPRAMC_1_PRCR1

P0_BO_DIS □

FT_DIS (0 -> 1) 0

Figure 4-62. Tresos Plugin snapshot for McuRamPRAMC_1_PRCR1 form.

4.9.18.47.2.1 McuFT_DIS (McuRamPRAMC_1_PRCR1)

PRAMC1_PRCR1[FT_DIS] field configuration.

Flow through disabled.

This field defines the AHB response on reads. The state of this field has no impact on the response latency on writes. This bit is cleared by hardware reset.

NOTE: Do not change the FT_DIS bit value while accessing system RAM. Relocate code programming

the FT_DIS bit to another memory area, e.g., local core memory.

0 RAM read data is passed directly to the system bus, incurring no additional latency

1 RAM read data is registered prior to returning on the system bus, incurring 1 extra cycle of latency

Note: Implementation Specific Parameter.

Table 4-368. Attribute McuP0BODIS (McuRamPRAMC_1_PRCR1) detailed description

Property	Value
Label	P0_BO_DIS
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	failse

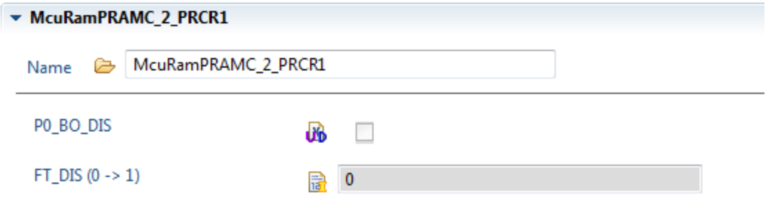
Table 4-369. Attribute McuFT_DIS (McuRamPRAMC_1_PRCR1) detailed description

Property	Value
Label	FT_DIS
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.47.3 Form McuRamPRAMC_2_PRCR1

The Platform RAM Configuration Register (PRAMC_2_PRCR1) is used to specify operation of the quad-port RAM controller's internal controller for SRAM bank 0. Note: Implementation Specific Parameter.

Is included by form : Form McuRam



McuRamPRAMC_2_PRCR1

Name: McuRamPRAMC_2_PRCR1

P0_BO_DIS □

FT_DIS (0 -> 1): 0

Figure 4-63. Tresos Plugin snapshot for McuRamPRAMC_2_PRCR1 form.

4.9.18.47.3.1 McuFT_DIS (McuRamPRAMC_2_PRCR1)

PRAMC2_PRCR1[FT_DIS] field configuration.

Flow through disabled.

This field defines the AHB response on reads. The state of this field has no impact on the response

latency on writes. This bit is cleared by hardware reset.

NOTE: Do not change the FT_DIS bit value while accessing system RAM. Relocate code programming

the FT_DIS bit to another memory area, e.g., local core memory.

0 RAM read data is passed directly to the system bus, incurring no additional latency

1 RAM read data is registered prior to returning on the system bus, incurring 1 extra cycle of latency

Note: Implementation Specific Parameter.

Table 4-370. Attribute McuP0BODIS (McuRamPRAMC_2_PRCR1) detailed description

Property	Value
Label	P0_BO_DIS
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

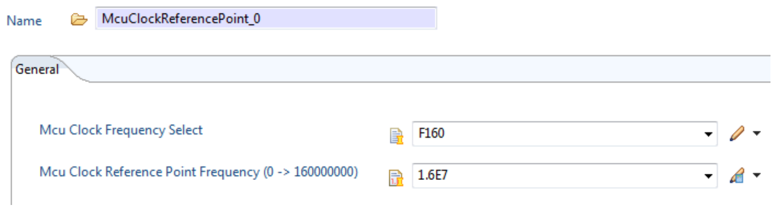
Table 4-371. Attribute McuFT_DIS (McuRamPRAMC_2_PRCR1) detailed description

Property	Value
Label	FT_DIS
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.18.48 Form McuClockReferencePoint

This container defines a reference point in the Mcu Clock tree. It defines the frequency which then can be used by other modules as an input value. Lower multiplictiy is 1, as even in the simpliest case (only one frequency is used), there is one frequency to be defined.

Is included by form : Form McuClockSettingConfig



Name McuClockReferencePoint_0

Mcu Clock Frequency Select F160

Mcu Clock Reference Point Frequency (0 -> 160000000) 1.6E7

Figure 4-64. Tresos Plugin snapshot for McuClockReferencePoint form.

4.9.18.48.1 McuClockFrequencySelect (McuClockReferencePoint)

List containing the available clock refference points.

Note: Implementation Specific Parameter.

Table 4-372. Attribute McuClockFrequencySelect (McuClockReferencePoint) detailed description

Property	Value
Label	Mcu Clock Frequency Select
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	F160
Range	SIRC

Table 4-372. Attribute McuClockFrequencySelect (McuClockReferencePoint) detailed description

Property	Value
	FIRC SXOSC FXOSC F160 S160 S80 FS80 S40 F40 F80 F20 CLKOUT1 ENET ENET_TIMER TRNG uSDHC PLL_Aux CLKOUT0 LIN0 SPI0 FLEXCAN0 EMIOS0 EMIOS1 EMIOS2 CUSTOM

4.9.18.48.2 McuClockReferencePointFrequency (McuClockReferencePoint)

This is the frequency for the specific instance of the McuClockReferencePoint container.

It shall be given in Hz.

Calculated value.

Table 4-373. Attribute McuClockReferencePointFrequency (McuClockReferencePoint) detailed description

Property	Value
Label	Mcu Clock Reference Point Frequency
Type	FLOAT
Origin	AUTOSAR_ECUC
Symbolic Name	false
Invalid	Range <=160000000 >=0

4.9.19 Form McuModeSettingConf

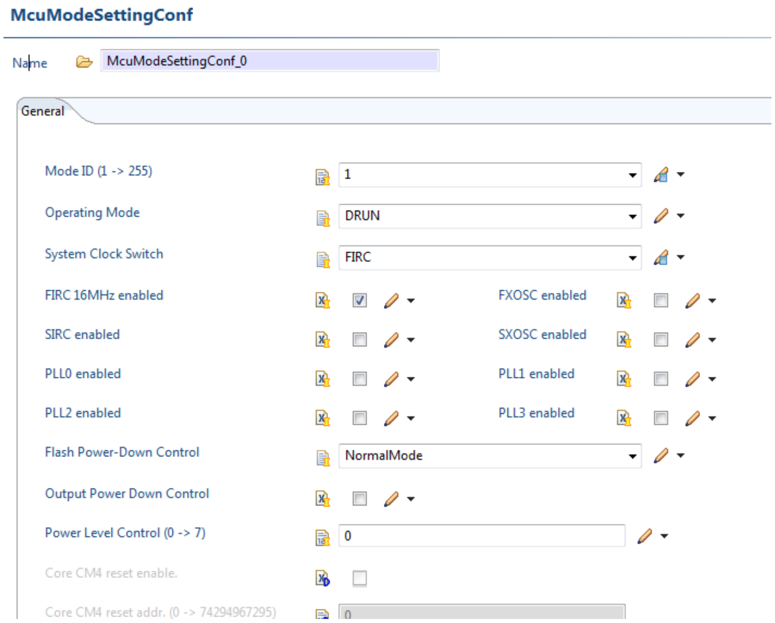
This container contains the configuration for the Mode setting of the MCU.

Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration

Included forms :

- Form McuCoreConfiguration
- Form McuModeSettingConfAfterWkp
- Form McuLpuConfiguration



McuModeSettingConf

Name McuModeSettingConf_0

General

Mode ID (1 -> 255) 1

Operating Mode DRUN

System Clock Switch FIRC

FIRC 16MHz enabled ☑ FXOSC enabled □

SIRC enabled □ SXOSC enabled □

PLL0 enabled □ PLL1 enabled □

PLL2 enabled □ PLL3 enabled □

Flash Power-Down Control NormalMode

Output Power Down Control □

Power Level Control (0 -> 7) 0

Core CM4 reset enable. □

Core CM4 reset addr. (0 -> 74294967295) 0

Figure 4-65. Tresos Plugin snapshot for McuModeSettingConf form.

4.9.19.1 McuMode (McuModeSettingConf)

This parameter shall represent the ID of the MCU mode.

Table 4-374. Attribute McuMode (McuModeSettingConf) detailed description

Property	Value
Label	Mode ID
Type	INTEGER
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-374. Attribute McuMode (McuModeSettingConf) detailed description (continued)

Property	Value
Symbolic Name	true
Invalid	Range <=255 >=0

4.9.19.2 McuPowerMode (McuModeSettingConf)

This parameter selects the Power Mode to be used.

For valid Mode transitions refers to "MC_ME Mode Diagram" from Reference Manual.

Note: Please ensure before entering STANDBY0 or LPU_RUN Mode, PMCDIG_MCR[LVD_IO_HI_REE] and PMCDIG_MCR[LVD_PD2_COLD_REE] bits are programmed to 0 (Eraata ERR010447).

Note: If STANDBY0 mode is chosen on Calypso 3M, please pay attention to errata ERR009139 (SRAM contents are not guaranteed in STANDBY Modes).

To enable SRAM retention in STANDBY do the step following (step1 and step2 were supported by MCU driver):

Step 1: Enable GPR_SLEEP_BIT prior to STANDBY entry (no SRAM access is permitted after this write)

Step 2: Perform the mode change to enter into STANBDY Mode

Step 3: On wakeup from STANDBY, clear the GPR_SLEEP_BIT before any access is made to the SRAM

Note: Implementation Specific Parameter.

Table 4-375. Attribute McuPowerMode (McuModeSettingConf) detailed description

Property	Value
Label	Operating Mode
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	DRUN
Range	RUN3

Table 4-375. Attribute McuPowerMode (McuModeSettingConf) detailed description

Property	Value
	RUN2 RUN1 RUN0 DRUN STOP0 STANDBY0 LPU_RUN SAFE RESET RESET_DEST

4.9.19.3 McuSystemClockSwitch (McuModeSettingConf)

System Clock Select. Configure the ME_mode_MC[SYSCLK] register field.

Provides the clock (divided or not) to the Core/part of Peripherals.

The system clock is either:

- Internal 16Mhz oscillator IRC

- External oscillator XOSC

- PLL0

Value extracted from Resource: MCU.CLockSelect.List

If the 4-40 MHz crystal osc. clock is selected as the system clock, XOSC must be on. f the system PLL clock is selected as the system clock, PLL0 must be on. Note: Implementation Specific Parameter.

Table 4-376. Attribute McuSystemClockSwitch (McuModeSettingConf) detailed description

Property	Value
Label	System Clock Switch
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.19.4 McuFircControl (McuModeSettingConf)

Fast 16 MHz internal RC oscillator control. Configure the ME_mode_MC[RCON] register field. This option should be used only for STBY0 states, else is by default configured with '1' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-377. Attribute McuFircControl (McuModeSettingConf) detailed description

Property	Value
Label	FIRC 16MHz enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.5 McuFXoscControl (McuModeSettingConf)

Fast XOSC oscillator control. Configure the ME_mode_MC[FXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-378. Attribute McuFXoscControl (McuModeSettingConf) detailed description

Property	Value
Label	FXOSC enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.6 McuSIrcOscControl (McuModeSettingConf)

Slow IRC oscillator control. Configure the ME_mode_MC[SIRCON] register field. Note: Implementation Specific Parameter.

Table 4-379. Attribute McuSIrcOscControl (McuModeSettingConf) detailed description

Property	Value
Label	SIRC enabled
Type	BOOLEAN

Table continues on the next page...

Table 4-379. Attribute McuSlrcOscControl (McuModeSettingConf) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.7 McuSXOscControl (McuModeSettingConf)

Slow XOSC oscillator control. Configure the ME_mode_MC[SXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-380. Attribute McuSXOscControl (McuModeSettingConf) detailed description

Property	Value
Label	SXOSC enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.8 McuPll0Control (McuModeSettingConf)

PLL0 oscillator control. Configure the ME_mode_MC[PLL0ON] register field. Note: Implementation Specific Parameter.

Table 4-381. Attribute McuPll0Control (McuModeSettingConf) detailed description

Property	Value
Label	PLL0 enabled
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.9 McuFlashPowerControl (McuModeSettingConf)

Code flash power-down control. Configure the ME_mode_MC[FLAON] register field. This option should be used only for DRUN/RUN0/RUN1/RUN2/RUN3/STOP0 states. In Standby mode the value is read only and set to PowerDown. In RESET and SAFE the value is read only and set to NormalMode. Note: Implementation Specific Parameter.

Table 4-382. Attribute McuFlashPowerControl (McuModeSettingConf) detailed description

Property	Value
Label	Flash Power-Down Control
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	NormalMode
Range	PowerDown NormalMode

4.9.19.10 McuOutputPowerDownControl (McuModeSettingConf)

I/O output power-down control Configure the ME_mode_MC[PDO] register field. This option should be used only for SAFE/STOP0 states, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-383. Attribute McuOutputPowerDownControl (McuModeSettingConf) detailed description

Property	Value
Label	Output Power Down Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.11 McuMainVoltageRegulatorControl (McuModeSettingConf)

Main voltage regulator control

Configure the ME_mode_MC[MVRON] register field.

This option should be configured only for STOP0 mode;

For STANDBY mode it is switched off by default and for the others is by default configured with '1' value and the field is read-only.

Note: Implementation Specific Parameter.

Table 4-384. Attribute McuMainVoltageRegulatorControl (McuModeSettingConf) detailed description

Property	Value
Label	Main Voltage Regulator Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.12 McuPowerLevelControl (McuModeSettingConf)

Power Level. These value indicates the relative power consumtion level of this mode with respect to that of other modes. This field is read only and set to 0 for STANDBY0, STOP0 an RESET modes. Note: Implementation Specific Parameter.

Table 4-385. Attribute McuPowerLevelControl (McuModeSettingConf) detailed description

Property	Value
Label	Power Level Control
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.19.13 McuFastTransition (McuModeSettingConf)

This option will enable to configure the ME_mode_MC register for current mode to have different settings after wakeup. Any modification of the mode configuration register of the currently selected mode will not be taken into account immediately but on the next request to enter this mode. Register ME_mode_MC will not be set if this option is true. This parameter is enabled only if "McuPowerMode" is STOP0 or STANDBY0 or LPU_RUN. Note: Implementation Specific Parameter.

Table 4-386. Attribute McuFastTransition (McuModeSettingConf) detailed description

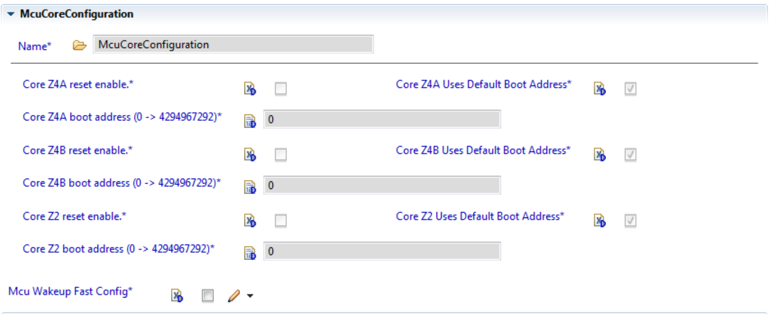
Property	Value
Label	Mcu Wakeup Fast Config
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.14 Form McuCoreConfiguration

This registers contains the boot address of COREs

Note: Implementation Specific Parameter.

Is included by form : Form McuModeSettingConf



McuCoreConfiguration

Name* McuCoreConfiguration

Core Z4A reset enable.* □ Core Z4A Uses Default Boot Address* ☑

Core Z4A boot address (0 -> 4294967292)* 0

Core Z4B reset enable.* □ Core Z4B Uses Default Boot Address* ☑

Core Z4B boot address (0 -> 4294967292)* 0

Core Z2 reset enable.* □ Core Z2 Uses Default Boot Address* ☑

Core Z2 boot address (0 -> 4294967292)* 0

Mcu Wakeup Fast Config* □

Figure 4-66. Tresos Plugin snapshot for McuCoreConfiguration form.

4.9.19.14.1 McuCoreCaddr1ResetEnable (McuCoreConfiguration)

Set this to TRUE to enable core Z4A Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-387. Attribute McuCoreCaddr1ResetEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4A reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.14.2 McuCoreZ4ADefaultBootAddrEnable (McuCoreConfiguration)

1 - Core Z4A Boot Address is not updated.

0 - Core Z4A Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-388. Attribute McuCoreZ4ADefaultBootAddrEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4A Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.14.3 McuCoreCaddr1BootAddr (McuCoreConfiguration)

Boot address for core Z4A after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

Table 4-389. Attribute McuCoreCaddr1BootAddr (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4A boot address
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.19.14.4 McuCoreCaddr2ResetEnable (McuCoreConfiguration)

Set this to TRUE to enable core Z4B Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-390. Attribute McuCoreCaddr2ResetEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4B reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.14.5 McuCoreZ4BDefaultBootAddrEnable (McuCoreConfiguration)

1 - Core Z4B Boot Address is not updated.

0 - Core Z4B Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-391. Attribute McuCoreZ4BDefaultBootAddrEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4B Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.14.6 McuCoreCaddr2BootAddr (McuCoreConfiguration)

Boot address for core Z4B after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

Table 4-392. Attribute McuCoreCaddr2BootAddr (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z4B boot address
Type	INTEGER

Table continues on the next page...

Table 4-392. Attribute McuCoreCaddr2BootAddr (McuCoreConfiguration) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	0

4.9.19.14.7 McuCoreCaddr3ResetEnable (McuCoreConfiguration)

Set this to TRUE to enable core Z2 Reset after the mode change

Note: Implementation Specific Parameter.

Table 4-393. Attribute McuCoreCaddr3ResetEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z2 reset enable.
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.14.8 McuCoreZ2DefaultBootAddrEnable (McuCoreConfiguration)

1 - Core Z2 Boot Address is not updated.

0 - Core Z2 Boot Address is updated.

Note: Implementation Specific Parameter.

Table 4-394. Attribute McuCoreZ2DefaultBootAddrEnable (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z2 Uses Default Boot Address
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.14.9 McuCoreCaddr3BootAddr (McuCoreConfiguration)

Boot address for core Z2 after the mode change.

The value from this field will be masked with 0xFFFFFFFC.

Note: Implementation Specific Parameter.

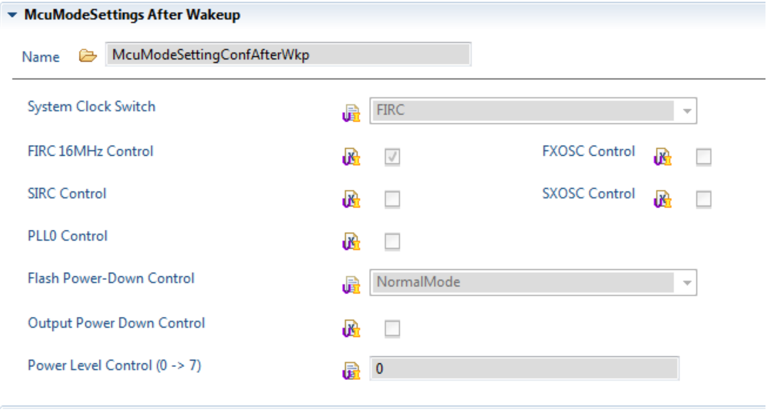
Table 4-395. Attribute McuCoreCaddr3BootAddr (McuCoreConfiguration) detailed description

Property	Value
Label	Core Z2 boot address
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.19.15 Form McuModeSettingConfAfterWkp

This container contains the configuration for the Mode setting of the MCU. These options are valid only if the selected mode is one of low power type (STANDBY0, LPU_RUN or STOP0). Note: Implementation Specific Parameter.

Is included by form : Form McuModeSettingConf



McuModeSettings After Wakeup

Name: McuModeSettingConfAfterWkp

System Clock Switch	FIRC		
FIRC 16MHz Control	☑	FXOSC Control	□
SIRC Control	□	SXOSC Control	□
PLL0 Control	□		
Flash Power-Down Control	NormalMode		
Output Power Down Control	□		
Power Level Control (0 -> 7)	0		

Figure 4-67. Tresos Plugin snapshot for McuModeSettingConfAfterWkp form.

4.9.19.15.1 McuSystemClockSwitch (McuModeSettingConfAfterWkp)

System Clock Select. Configure the ME_mode_MC[SYSCLK] register field.

Provides the clock (divided or not) to the Core/part of Peripherals.

The system clock is either:

- Internal 16Mhz oscillator IRC

- External oscillator XOSC

- PLL0

Value extracted from Resource: MCU.CLockSelect.List

If the 4-40 MHz crystal osc. clock is selected as the system clock, XOSC must be on. f the system PLL clock is selected as the system clock, PLL0 must be on. Note: Implementation Specific Parameter.

Table 4-396. Attribute McuSystemClockSwitch (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	System Clock Switch
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.19.15.2 McuFircControl (McuModeSettingConfAfterWkp)

Fast 16 MHz internal RC oscillator control. Configure the ME_mode_MC[RCON] register field. This option should be used only for STBY0 states, else is by default configured with '1' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-397. Attribute McuFircControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	FIRC 16MHz Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.15.3 McuFXoscControl (McuModeSettingConfAfterWkp)

Fast XOSC oscillator control. Configure the ME_mode_MC[FXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-398. Attribute McuFXoscControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	FXOSC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.15.4 McuSIrcOscControl (McuModeSettingConfAfterWkp)

Slow IRC oscillator control. Configure the ME_mode_MC[SIRCON] register field. Note: Implementation Specific Parameter.

Table 4-399. Attribute McuSIrcOscControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	SIRC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.15.5 McuSXOscControl (McuModeSettingConfAfterWkp)

Slow XOSC oscillator control. Configure the ME_mode_MC[SXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-400. Attribute McuSXOscControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	SXOSC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.15.6 McuPll0Control (McuModeSettingConfAfterWkp)

PLL0 oscillator control. Configure the ME_mode_MC[PLL0ON] register field. Note: Implementation Specific Parameter.

Table 4-401. Attribute McuPll0Control (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	PLL0 Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.15.7 McuFlashPowerControl (McuModeSettingConfAfterWkp)

Code flash power-down control. Configure the ME_mode_MC[CFLAON] register field. Note: Implementation Specific Parameter.

Table 4-402. Attribute McuFlashPowerControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	Flash Power-Down Control
Type	ENUMERATION
Origin	Custom
Symbolic Name	false
Default	NormalMode
Range	PowerDown NormalMode

4.9.19.15.8 McuOutputPowerDownControl (McuModeSettingConfAfterWkp)

I/O output power-down control Configure the ME_mode_MC[PDO] register field. This option should be used only for SAFE/STOP0 states, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-403. Attribute McuOutputPowerDownControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	Output Power Down Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.15.9 McuPowerLevelControl (McuModeSettingConfAfterWkp)

Power Level. These value indicates the relative power consumtion level of this mode with respect to that of other modes. Note: Implementation Specific Parameter.

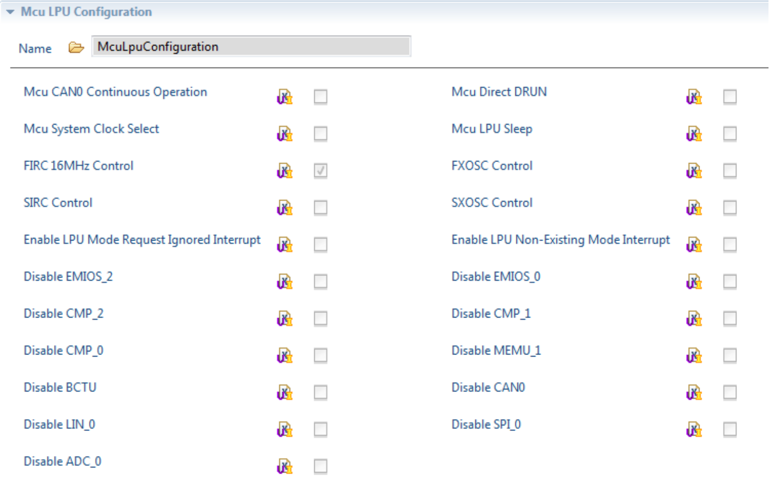
Table 4-404. Attribute McuPowerLevelControl (McuModeSettingConfAfterWkp) detailed description

Property	Value
Label	Power Level Control
Type	INTEGER
Origin	Custom
Symbolic Name	false
Default	0

4.9.19.16 Form McuLpuConfiguration

This container contains the configuration for the LPU module. These options are valid only if the selected mode is LPU_RUN. Note: Implementation Specific Parameter.

Is included by form : Form McuModeSettingConf



Mcu LPU Configuration

Name McuLpuConfiguration

Mcu CAN0 Continuous Operation	☐	Mcu Direct DRUN	☐
Mcu System Clock Select	☐	Mcu LPU Sleep	☐
FIRC 16MHz Control	☑	FXOSC Control	☐
SIRC Control	☐	SXOSC Control	☐
Enable LPU Mode Request Ignored Interrupt	☐	Enable LPU Non-Existing Mode Interrupt	☐
Disable EMIOS_2	☐	Disable EMIOS_0	☐
Disable CMP_2	☐	Disable CMP_1	☐
Disable CMP_0	☐	Disable MEMU_1	☐
Disable BCTU	☐	Disable CAN0	☐
Disable LIN_0	☐	Disable SPI_0	☐
Disable ADC_0	☐		

Figure 4-68. Tresos Plugin snapshot for McuLpuConfiguration form.

4.9.19.16.1 McuCAN0ContinuousOperation (McuLpuConfiguration)

CAN0 continuous operation across LPU modes. CAN CHI frequency will switch from FS80 to LPU_SYS_CLK which will require reconfiguration thus inhibiting continuous operation across LPU modes.

Configure the LPU_mode_CF[CAN0_CONT] register field.

0 Continuous CAN0 operation is not supported

1 Continuous CAN0 operation is supported. CAN0 needs to be operated by selected FXOSC as the source of its CHI clock by configuring MC_CGM_ACx

This option should be used only for LPU_RUN state, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-405. Attribute McuCAN0ContinuousOperation (McuLpuConfiguration) detailed description

Property	Value
Label	Mcu CAN0 Continuous Operation
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.2 McuDirectDrun (McuLpuConfiguration)

Transition to DRUN on LPU_STANDBY exit.

Configure the LPU_mode_CF[DIRECT_DRUN] register field.

0 Switch to LPU_RUN on exit from LPU_STANDBY

1 Switch to DRUN on exit from LPU_STANDBY

This option should be used only for LPU_RUN state, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-406. Attribute McuDirectDrun (McuLpuConfiguration) detailed description

Property	Value
Label	Mcu Direct DRUN
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.3 McuSystemClockSelect (McuLpuConfiguration)

System clock selection in LPU RUN mode.

Configure the LPU_mode_CF[SYS_CLK_SEL] register field.

0 Select FIRC clock source

1 Select FXOSC clock source

This option should be used only for LPU_RUN state, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-407. Attribute McuSystemClockSelect (McuLpuConfiguration) detailed description

Property	Value
Label	Mcu System Clock Select
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.4 McuLpuSleep (McuLpuConfiguration)

Core sleep request during LPU RUN.

Configure the LPU_mode_CF[LPU_SLEEP] register field.

0 No core sleep request

1 Core sleep request

This option should be used only for LPU_RUN state, else is by default configured with '0' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-408. Attribute McuLpuSleep (McuLpuConfiguration) detailed description

Property	Value
Label	Mcu LPU Sleep
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.5 McuFircControl (McuLpuConfiguration)

Fast 16 MHz internal RC oscillator control. Configure the LPU_mode_CF[FIRC_ON] register field. This option should be used only for LPU_STOP and LPU_STANDBY states, else is by default configured with '1' value and the field is read-only. Note: Implementation Specific Parameter.

Table 4-409. Attribute McuFircControl (McuLpuConfiguration) detailed description

Property	Value
Label	FIRC 16MHz Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	true

4.9.19.16.6 McuFXoscControl (McuLpuConfiguration)

Fast XOSC oscillator control. Configure the LPU_mode_CF[FXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-410. Attribute McuFXoscControl (McuLpuConfiguration) detailed description

Property	Value
Label	FXOSC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.7 McuSIrcOscControl (McuLpuConfiguration)

Slow IRC oscillator control. Configure the LPU_mode_CF[SIRCON] register field. Note: Implementation Specific Parameter.

Table 4-411. Attribute McuSIrcOscControl (McuLpuConfiguration) detailed description

Property	Value
Label	SIRC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.8 McuSXOscControl (McuLpuConfiguration)

Slow XOSC oscillator control. Configure the LPU_mode_CF[SXOSCON] register field. Note: Implementation Specific Parameter.

Table 4-412. Attribute McuSXOscControl (McuLpuConfiguration) detailed description

Property	Value
Label	SXOSC Control
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.9 McuLpuMRIGInterruptEnable (McuLpuConfiguration)

LPU mode request ignored interrupt enable. Configure the LPU_ICR[LPU_MRIG_E] register field. Note: Implementation Specific Parameter.

Table 4-413. Attribute McuLpuMRIGInterruptEnable (McuLpuConfiguration) detailed description

Property	Value
Label	Enable LPU Mode Request Ignored Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.10 McuLpuNEMInterruptEnable (McuLpuConfiguration)

LPU non-exist mode interrupt enable. Configure the LPU_ICR[LPU_NEM_E] register field. Note: Implementation Specific Parameter.

Table 4-414. Attribute McuLpuNEMInterruptEnable (McuLpuConfiguration) detailed description

Property	Value
Label	Enable LPU Non-Existing Mode Interrupt
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.11 McuDisableEmios2 (McuLpuConfiguration)

eMIOS_2 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that eMIOS_2 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[eMIOS_2] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-415. Attribute McuDisableEmios2 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable EMIOS_2
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.12 McuDisableEmios0 (McuLpuConfiguration)

eMIOS_0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that eMIOS_0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[eMIOS_0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-416. Attribute McuDisableEmios0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable EMIOS_0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.13 McuDisableCmp2 (McuLpuConfiguration)

CMP_2 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that CMP_2 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[CMP_2] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-417. Attribute McuDisableCmp2 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable CMP_2
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.14 McuDisableCmp1 (McuLpuConfiguration)

CMP_1 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that CMP_1 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[CMP_1] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-418. Attribute McuDisableCmp1 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable CMP_1
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.15 McuDisableCmp0 (McuLpuConfiguration)

CMP_0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that CMP_0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[CMP_0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-419. Attribute McuDisableCmp0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable CMP_0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.16 McuDisableMemu1 (McuLpuConfiguration)

MEMU_1 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that MEMU_1 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[MEMU_1] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-420. Attribute McuDisableMemu1 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable MEMU_1
Type	BOOLEAN

Table continues on the next page...

Table 4-420. Attribute McuDisableMemu1 (McuLpuConfiguration) detailed description (continued)

Property	Value
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.17 McuDisableBctu (McuLpuConfiguration)

BCTU is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that BCTU will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[BCTU] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-421. Attribute McuDisableBctu (McuLpuConfiguration) detailed description

Property	Value
Label	Disable BCTU
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.18 McuDisableCan0 (McuLpuConfiguration)

CAN0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that CAN0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[CAN0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-422. Attribute McuDisableCan0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable CAN0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.19 McuDisableLin0 (McuLpuConfiguration)

LIN_0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that LIN_0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[LIN_0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-423. Attribute McuDisableLin0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable LIN_0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.20 McuDisableSpi0 (McuLpuConfiguration)

SPI_0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that SPI_0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[SPI_0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

Table 4-424. Attribute McuDisableSpi0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable SPI_0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.19.16.21 McuDisableAdc0 (McuLpuConfiguration)

ADC_0 is disabled in RUN mode using ME_PCTL, it is suggested to configure MDIS to 1 in LPU modes as by default the clock to that ADC_0 will be enabled while transitioning to LPU mode.

Configure the LPU_MDIS[ADC_0] register field.

0 Module clock is enabled in LPU_RUN or LPU_SLEEP mode.

1 Module clock is disabled in LPU_RUN or LPU_SLEEP mode.

Note: Implementation Specific Parameter.

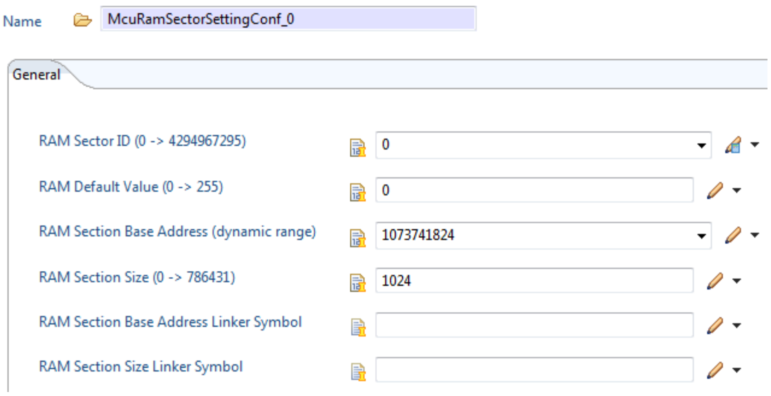
Table 4-425. Attribute McuDisableAdc0 (McuLpuConfiguration) detailed description

Property	Value
Label	Disable ADC_0
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.20 Form McuRamSectorSettingConf

This container contains the configuration for the RAM Sector setting.

Is included by form : Form McuModuleConfiguration



Name McuRamSectorSettingConf_0

General

RAM Sector ID (0 -> 4294967295): 0

RAM Default Value (0 -> 255): 0

RAM Section Base Address (dynamic range): 1073741824

RAM Section Size (0 -> 786431): 1024

RAM Section Base Address Linker Symbol:

RAM Section Size Linker Symbol:

Figure 4-69. Tresos Plugin snapshot for McuRamSectorSettingConf form.

4.9.20.1 McuRamSectorId (McuRamSectorSettingConf)

This parameter shall represent the ID of the MCU RAM Sector configuration.

Table 4-426. Attribute McuRamSectorId (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Sector ID
Type	INTEGER
Origin	Custom
Symbolic Name	true

4.9.20.2 McuRamDefaultValue (McuRamSectorSettingConf)

This parameter shall represent the Data pre-setting to be initialized. Default value is 0xbababababa. Note: Implementation Specific Parameter.

Table 4-427. Attribute McuRamDefaultValue (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Default Value
Type	INTEGER
Origin	AUTOSAR_ECUC

Table continues on the next page...

Table 4-427. Attribute McuRamDefaultValue (McuRamSectorSettingConf) detailed description (continued)

Property	Value
Symbolic Name	false
Default	0
Invalid	Range <=255 >=0

4.9.20.3 McuRamSectionBaseAddress (McuRamSectorSettingConf)

This parameter represents the RAM section base address. The address must be aligned to 4 bytes. Note: Implementation Specific Parameter.

Table 4-428. Attribute McuRamSectionBaseAddress (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Section Base Address
Type	INTEGER
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1073741824
Invalid	Range <ecu:get('Mcu.McuModuleConfiguration.McuRamSectorSettingConf.McuRamSectionBaseAddress.high') >=ecu:get('Mcu.McuModuleConfiguration.McuRamSectorSettingConf.McuRamSectionBaseAddress.low')

4.9.20.4 McuRamSectionSize (McuRamSectorSettingConf)

This parameter represents the RAM section size in bytes.

Note: Implementation Specific Parameter.

Table 4-429. Attribute McuRamSectionSize (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Section Size
Type	INTEGER

Table continues on the next page...

Table 4-429. Attribute McuRamSectionSize (McuRamSectorSettingConf) detailed description (continued)

Property	Value
Origin	AUTOSAR_ECUC
Symbolic Name	false
Default	1024
Invalid	Range <=786431 >=0

4.9.20.5 McuRamSectionBaseAddrLinkerSym (McuRamSectorSettingConf)

This parameter represents the RAM section base address. The address must be aligned to 4 bytes. If this parameter is empty, then the integer values from "McuRamSectionBaseAddress" will be used. Note: Implementation Specific Parameter.

Table 4-430. Attribute McuRamSectionBaseAddrLinkerSym (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Section Base Address Linker Symbol
Type	STRING
Origin	Custom
Symbolic Name	false
Default	

4.9.20.6 McuRamSectionSizeLinkerSym (McuRamSectorSettingConf)

This parameter represents the RAM section size in bytes. The size must be multiple of 4. If this parameter is empty, then the integer values from "McuRamSectionSize" will be used. Note: Implementation Specific Parameter.

Table 4-431. Attribute McuRamSectionSizeLinkerSym (McuRamSectorSettingConf) detailed description

Property	Value
Label	RAM Section Size Linker Symbol
Type	STRING
Origin	Custom

Table continues on the next page...

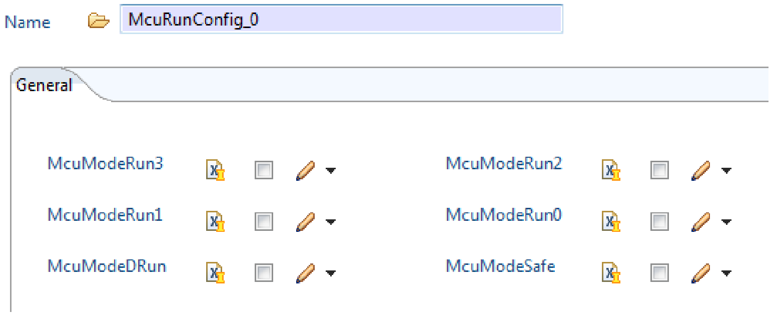
Table 4-431. Attribute McuRamSectionSizeLinkerSym (McuRamSectorSettingConf) detailed description (continued)

Property	Value
Symbolic Name	false
Default	

4.9.21 Form McuRunConfig

This container is for Peripheral Run Configurations. Configure the ME_RUN_PC0..7 registers. Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration



Name: McuRunConfig_0

Figure 4-70. Tresos Plugin snapshot for McuRunConfig form.

4.9.21.1 McuModeRun3 (McuRunConfig)

Check if you want to enable the peripheral in RUN3 mode. This parameter can be used only if ../../../McuEnableMode/McuModeRun3 is ON. Note: Implementation Specific Parameter.

Table 4-432. Attribute McuModeRun3 (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.21.2 McuModeRun2 (McuRunConfig)

Check if you want to enable the peripheral in RUN2 mode. This parameter can be used only if ../../../McuEnableMode/McuModeRun2 is ON. Note: Implementation Specific Parameter.

Table 4-433. Attribute McuModeRun2 (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.21.3 McuModeRun1 (McuRunConfig)

Check if you want to enable the peripheral in RUN1 mode. This parameter can be used only if ../../../McuEnableMode/McuModeRun1 is ON. Note: Implementation Specific Parameter.

Table 4-434. Attribute McuModeRun1 (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.21.4 McuModeRun0 (McuRunConfig)

Check if you want to enable the peripheral in RUN0 mode. Note: Implementation Specific Parameter.

Table 4-435. Attribute McuModeRun0 (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.21.5 McuModeDRun (McuRunConfig)

Check if you want to enable the peripheral in DRUN mode. Note: Implementation Specific Parameter.

Table 4-436. Attribute McuModeDRun (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.21.6 McuModeSafe (McuRunConfig)

Check if you want to enable the peripheral in SAFE mode. Note: Implementation Specific Parameter.

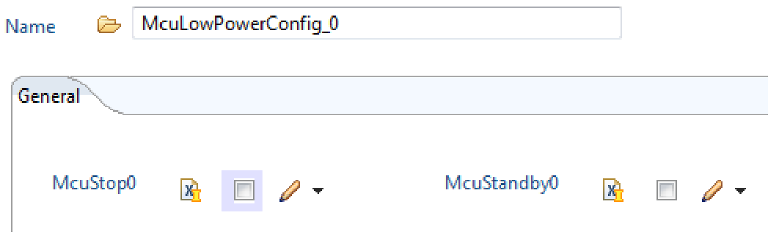
Table 4-437. Attribute McuModeSafe (McuRunConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.22 Form McuLowPowerConfig

This container is for Peripheral Low-Power Configurations. Configure the ME_LP_PC0..7 registers. Note: Implementation Specific Parameter.

Is included by form : Form McuModuleConfiguration



Name McuLowPowerConfig_0

Figure 4-71. Tresos Plugin snapshot for McuLowPowerConfig form.

4.9.22.1 McuStop0 (McuLowPowerConfig)

Check if you want to enable the peripheral in STOP mode. Note: Implementation Specific Parameter.

Table 4-438. Attribute McuStop0 (McuLowPowerConfig) detailed description

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.22.2 McuStandby0 (McuLowPowerConfig)

Check if you want to enable the peripheral in STANDBY0 mode. Note: Implementation Specific Parameter.

Table 4-439. Attribute McuStandby0 (McuLowPowerConfig) detailed description

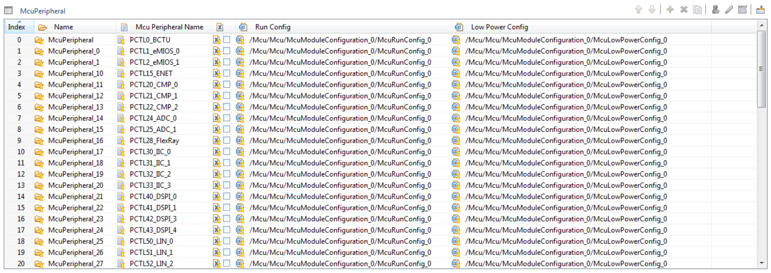
Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.23 Form McuPeripheral

This contains the combination for current peripheral in Run and LowPower Mode.

Note: Implementation Specific Container.

Is included by form : Form McuModuleConfiguration



McuPeripheral

Index	Name	Mcu Peripheral Name		Run Config	Low Power Config
0	McuPeripheral	PCTL0_BCTU	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
1	McuPeripheral_0	PCTL1_eMIOS_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
2	McuPeripheral_1	PCTL2_eMIOS_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
3	McuPeripheral_10	PCTL15_ENET	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
4	McuPeripheral_11	PCTL20_CMP_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
5	McuPeripheral_12	PCTL21_CMP_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
6	McuPeripheral_13	PCTL22_CMP_2	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
7	McuPeripheral_14	PCTL24_ADC_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
8	McuPeripheral_15	PCTL25_ADC_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
9	McuPeripheral_16	PCTL28_FlexRay	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
10	McuPeripheral_17	PCTL30_IIC_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
11	McuPeripheral_18	PCTL31_IIC_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
12	McuPeripheral_19	PCTL32_IIC_2	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
13	McuPeripheral_20	PCTL33_IIC_3	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
14	McuPeripheral_21	PCTL40_DSPI_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
15	McuPeripheral_22	PCTL41_DSPI_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
16	McuPeripheral_23	PCTL42_DSPI_3	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
17	McuPeripheral_24	PCTL43_DSPI_4	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
18	McuPeripheral_25	PCTL50_LIN_0	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
19	McuPeripheral_26	PCTL51_LIN_1	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0
20	McuPeripheral_27	PCTL52_LIN_2	☐	/Mcu/Mcu/McuModuleConfiguration_0/McuRunConfig_0	/Mcu/Mcu/McuModuleConfiguration_0/McuLowPowerConfig_0

Figure 4-72. Tresos Plugin snapshot for McuPeripheral form.

4.9.23.1 McuPerName (McuPeripheral)

This is the name of the peripheral.

Note: Implementation Specific Parameter.

Table 4-440. Attribute McuPerName (McuPeripheral) detailed description

Property	Value
Label	Mcu Peripheral Name
Type	ENUMERATION
Origin	Custom
Symbolic Name	false

4.9.23.2 McuPerDBGConfig (McuPeripheral)

Sets MC_ME_PCTLx[DBG_F] bit. Peripheral control in debug mode: This bit controls the state of the peripheral in debug mode.

This feature is useful to freeze the peripheral state while entering debug.

0 - Peripheral state depends on RUN_CFG/LP_CFG bits and the chip mode.

1 - Peripheral is frozen if not already frozen in chip modes.

Note: Implementation Specific Parameter.

Table 4-441. Attribute McuPerDBGConfig (McuPeripheral) detailed description

Property	Value
Label	Peripheral control in debug mode

Table continues on the next page...

Table 4-441. Attribute McuPerDBGConfig (McuPeripheral) detailed description (continued)

Property	Value
Type	BOOLEAN
Origin	Custom
Symbolic Name	false
Default	false

4.9.23.3 McuPerRunConfig (McuPeripheral)

The configuration selected for this peripheral in Run Mode. Note: Implementation Specific Parameter.

Table 4-442. Attribute McuPerRunConfig (McuPeripheral) detailed description

Property	Value
Label	Run Modes Configuration
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom

4.9.23.4 McuPerLowPwrConfig (McuPeripheral)

The configuration selected for this peripheral in Low-Power Mode. Note: Implementation Specific Parameter.

Table 4-443. Attribute McuPerLowPwrConfig (McuPeripheral) detailed description

Property	Value
Label	Non-Run Modes Configuration
Type	SYMBOLIC-NAME-REFERENCE
Origin	Custom

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.