User Manual

for S32K3 PLATFORM Driver

Document Number: UM34PLATFORMASR21-11 Rev0000R3.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	31.03.2023	NXP RTD Team	S32K3 Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 3.0.0

Chapter 2

Introduction

This User Manual describes the NXP Semiconductor PLATFORM driver for S32K3. The PLATFORM driver configuration parameters and deviations from the specification are described in PLATFORM Driver chapter of this document. PLATFORM driver requirements and APIs are vendor-specific.

2.1 Supported Derivatives

The software described in this document is intended to be used with the following microcontroller devices of NXP Semiconductors:

- s32k310_mqfp100
- s32k310_lqfp48
- s32k311_mqfp100 / MWCT2015S_mqfp100
- s32k311_lqfp48
- s32k312_mqfp100 / MWCT2016S_mqfp100
- s32k312_mqfp172 / MWCT2016S_mqfp172
- s32k314_mqfp172
- s32k314_mapbga257
- s32k322_mqfp100 / MWCT2D16S_mqfp100
- s32k322_mqfp172 / MWCT2D16S_mqfp172

- s32k324_mqfp172 / MWCT2D17S_mqfp172
- s32k324_mapbga257
- s32k341_mqfp100
- s32k341_mqfp172
- s32k342_mqfp100
- s32k342_mqfp172
- s32k344_mqfp172
- s32k344_mapbga257
- s32k394_mapbga289
- s32k396_mapbga289
- s32k358_mqfp172
- s32k358_mapbga289
- s32k328_mqfp172
- s32k328_mapbga289
- s32k338_mqfp172
- s32k338_mapbga289
- s32k348_mqfp172
- s32k348_mapbga289
- s32m274_lqfp64
- s32m276_lqfp64

All of the above microcontroller devices are collectively named as S32K3.

Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power.

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** style: Used for important terms, notes and warnings.
- *Italic* style: 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.

Warning

This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
C/CPP	C and C++ Source Code
CS	Chip Select
CTU	Cross Trigger Unit
DEM	Diagnostic Event Manager
DET	Development Error Tracer
DMA	Direct Memory Access
ECU	Electronic Control Unit
FIFO	First In First Out
IRQ	Interrupt request
LSB	Least Signifigant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
SIU	Systems Integration Unit
SWS	Software Specification
VLE	Variable Length Encoding
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	General Specification of Basic Software Modules	AUTOSAR Release R21-11
2	Specification of Communication Stack Types	AUTOSAR Release R21-11
3	Specification of Compiler Abstraction	AUTOSAR Release R21-11
4	Specification of Platform Types	AUTOSAR Release R21-11
5	Specification of Standard Types	AUTOSAR Release R21-11
6	S32K3xx Reference Manual	S32K3xx Reference Manual,Rev.6, Draft B, 01/2023
7	S32K39 and S32K37 Reference Manual	Rev. 2 Draft A, 11/2022
8	S32M27x Reference Manual	Rev.2, Draft A, 02/2023
9	S32K3xx Datasheet	S32K3xx Data Sheet, Rev. 6, 11/202
10	S32K396 Datasheet	Rev. 1.1 — 08/2022
11	S32M2xx Datasheet	S32M2xx Data Sheet, Rev. 2 RC — 12/2022
12	S32K311 Errata	S32K311_0P98C Mask Set Errata, Rev. 6/March/2023, 3/2023

#	Title	Version
13	S32K312 Errata	Mask Set Errata for Mask 0P09C, Rev. 25/April/2022
14	S32K342 Errata	Mask Set Errata for Mask 0P97C, Rev. 10, 11/2022
15	S32K3x4 Errata	Mask Set Errata for Mask 0P55A/1P55A, Rev. 14/↩ Oct/2022
16	S32K358 Errata	S32K358_0P14E Mask Set Errata – Rev. 28, 9/2022
17	S32K396 Errata	S32K396_0P40E Mask Set Errata, Rev. DEC2022, 12/2022

Chapter 3

Driver

3.1 Requirements

PLATFORM is a complex driver, so there are no AUTOSAR requirements specific to this module. For the S32K3 platform, the PLATFORM module configures the interrupt controller, MCM, MSCM and INTM functionalities. It has vendor-specific requirements and implementation.

3.2 Driver Design Summary

The PLATFORM driver configures platform specific settings, managing the interrupt requests and other system wide settings as defined in each hardware implementation.

Interrupt Controller

The configuration contains the list of interrupt requests, as defined per each platform; the application can enable the interrupts and set priorities. There is one Interrupt Controller belonging to each M7 core.

Generic Interrupt Settings

The configuration contains the list of interrupt requests, as defined per each platform; the application can set interrupt routing to each M7 cores. The IsrHandler can be defined here or installed/updated at runtime via API.

Note

> The handler installation works only if the interrupt vector table resides in RAM; the default implementation for startup/linker files provided by NXP enables this functionality.

System Settings

On S32K3, PLATFORM driver allows the configuration of core-related interrupt requests, as defined in the implementation of MCM module, as well as other system specific settings (e.g. AHB slave access priority)

Interrupt Monitors

The PLATFORM driver provides configuration support, as well as APIs that can be called at runtime, in order to initialize the interrupt monitors and retrieve the status for the monitored IRQs.

Below you can find the descriptions for each file present in the Platform module:

File Name	File Type	Description
nvic.h	Stub file. Must be replaced by all integrators.	This file is a stub. This file include set priority grouping, enable , disable, set priority interrupt.
sys_init.h	Stub file. Must be replaced by all integrators.	this file is a stub. This file include some functions Function used to disable the interrupt number id. Function used to enable the interrupt number id and set up the priority. Function used to register the interrupt handler in the interrupt vectors. Function used to enable all interrupts. Function used to disable all interrupts. Function used to initiatialize clocks, system clock is system Pll 120 MHz. Function used to enter halt mode. Function used to enter stop mode. Function used to provide the CoreID to EUnit.
system.h	Stub file. Must be replaced by all integrators.	This file is a stub. This file include define some macros SCB Interrupt Control State Register Definitions.
core_↩ specific.h	Stub file. Must be replaced by all integrators.	This file is a stub. This file necessary for mpu memory region configuration.
nvic.c	Stub file. Must be replaced by all integrators.	This file is a stub. Set Priority Grouping. The function sets the priority grouping field using the required unlock sequence. The parameter PriorityGroup is assigned to the field SCB->AIRCR [10:8] PRIGROUP field.
sys_init.c	Stub file. Must be replaced by all integrators.	This file is a stub. This file include some functions need to initialize the clock.
system.c	Stub file. Must be replaced by all integrators.	This file is a stub. This file include some function. Function used to enter to supervisor mode. Check if it is needed to switch to supervisor mode and make the switch.

File Name	File Type	Description
startup.c	Stub file. Must be replaced by all integrators.	This file is a stub. This file include sone functions necessary initializations for RAM. Copy the vector table from ROM to RAM. Copy initialized data from ROM to RAM. Copy code that should reside in RAM from ROM. Clear the zero-initialized data section.
exceptions.c	Stub file. Must be replaced by all integrators.	This file is a stub. This file necessary handler exception when running application.
startup_↩ cm7.s	Stub file. Must be replaced by all integrators.	This file is a stub. This file need to run before initiatial application, it start-up code shall initialize the base addresses for interrupt and trap vector tables.
Vector_↩ Table.s	Stub file. Must be replaced by all integrators.	This file is a stub. This file necessary initializations for vector table.

Linker files The provided linker file is a demo(reference) code, delivered for a single derivative (master part) common for all other derivatives.

- In order to benefit of the maximum available memory range, the integrators need to modify the linker file according to its specific derivatives.

- For example please refer to the Reference Manual attached memory map spreadsheet, change the memory section size in linker file according to the memory range for each derivatives.

3.3 Hardware Resources

#	Hardware IP	Description
1	S32 NVIC	ARM V7 Nested Vectored Interrupt Controller
2	MSCM	Miscellaneous System Control Module
3	MSM	Miscellaneous Control Module
4	INTM	Interrupt Monitor

3.4 Deviations from Requirements

The driver deviates from the Platform Driver Software Specification in some places.

The table Status Column Description identifies the requirements that are not fully implemented, implemented differently, or out of scope for the Platform Driver.

The table Platform requirements deviations provides the "Status" column description.

Term	Definition
N/S	Not In Scope
N/F	Not Fully Implemented
N/I	Not Implemented

3.4.0.0.1 Status Column Description

Requirement	Status	Description	Notes
Platform_031	N/S	An enumeration, named Platform_↩ GlobalStateType, shall expose the driver internal states: initialized/uninitialized.	Platform driver only implements registering and enabling/disabling interrupts, so this requirement Platform_031 is not necessary in platform driver. Please refer this ticket AAI-928 for details

3.4.0.0.2 Platform Requirements Deviations

3.5 Driver Limitations

The PLATFORM driver software have some following limitations for RTD S32K3 :

- Only one precompile configuration variant supported in the configuration tool
- MISRA violations not fixed/commented in stub file relate to startup
- Derivatives with low ram like S32K311, S32K310, S32M276, S32M274 only support linker to executing code on flash

3.6 Driver usage and configuration tips

Platform driver does not support startup code and linker script, but it contains a sample for startup and linker. It also contains MPU and Cache Initialization, please note that MPU and Cache enablement in startup code is just demo code, user can find detail about MPU support in the Platform driver, and Cache support in MCL driver. Here is some samples for configuration we can custom for startup:

- User can define processor -DD_CACHE_ENABLE and -DI_CACHE_ENABLE to enable Dcache and Icache at startup code.
- User can enable MPU default configuration from startup code by using preprocessor -DMPU_ENABLE.
- MPU need to be enabled prior to Cache enablement, make sure ENABLE_MPU and D_CACHE_ENABLE I_CACHE_ENABLE are defined together.
- The PLATFORM driver provides sample linkers that execute code on ram and flash:

  - RAM execution: Entire code and data will be downloaded directly to SRAM. This profile should not be fit for production but it's useful for testing, for example, faster loading image, does not reduce the life cycle of flash, has a lower wait state than flash, user can experiment with possible bricking experimental(error configuration on jtag pin, mcu clock, security feature, reset loop . . .). The disadvantage of this profile is when destructive reset or POR happens, SoC will lose it's own code and data on ram.
  - FLASH execution: Entire image will be loaded to internal/external flash. Each section of the image will be initialized by startup (demo) after the SoC boot properly. Users can refer to the startup folder in Platform to create a startup by themselves.

- There is a small amount of SRAM (check RM for details specific to each device) that is available when the MCU is in standby mode. This memory region allows to store critical data or important code that needs to be available when the MCU wakes from standby mode.
  - The data stored in the standby SRAM memory sourced by the standby domain is retained when the MCU is in standby mode and available after the wakeup. But the data in SRAM sourced by Run domain is not available and it needs to be initialized after the wakeup to avoid ECC errors. Platform offers a demo for Startup, it will detect the status of previous POR event to decide whether to initialize this data section or not.
  - Before going to standby mode, you standby section got cached, make sure to clean cache before going to standby mode to avoid losing data.
  - Variables placed in the .standby_data section should not be initialized with a value, similar to .bss section but will be retained when exiting standby mode.

The PLATFORM driver should generally be initialized before calling other software that requires platform specific setup, like interrupts configuration.

The PLATFORM is working on an external assumption that the Vector table is already set up by the startup. The Platform driver will initialize based on the current set up of the VTOR register. The PLATFORM does not set up the VTOR table for the following :

- It should be setup by startup. Changing VTOR during runtime can result unexpected behavior.
- The system exception should not change when the device is ready in use.
- To install IRQ handlers, the vector table should be moved to RAM.

3.6.1 Initialization

The driver configuration contains a list of all implemented interrupt requests, with the associated settings. Besides that, it exposes information about all the configurable system-level settings, as well as interrupt monitors (if available). After generating the configuration structure from either Tresos or S32 Configuration Tool, the *Platform_Init* function should be called in order to apply the settings at NVIC level (also MSCM interrupt-to-core routing, if available).
Similarly, at the IP layer, the *IntCtrl_Ip_Init* function configures the entire list of configured interrupts at once, based on a structure generated by the tools.

3.6.2 Handling the interrupts

If specific IRQs need to be configured alone, the dedicated API can be called to enable/disable, or set the priority for a single interrupt request (*Platform_SetIrq*, *Platform_SetIrqPriority*, and their equivalent at the IntCtrl_Ip level). The user can also optionally overwrite the default interrupt handler, by calling *Platform_InstallHandler* (or *IntCtrl_Ip_InstallHandler* at IP level).
The parameter for all interrupt-related APIs that identifies the interrupt request being handled is an enumeration called *IRQn_Type*, defined for each SoC in the platform header file.

3.6.3 Handling the interrupt monitors

If individual interrupt monitors need to be handled besides the configuration structure, the driver provides dedicated API both at PLATFORM HLD and INTM IP levels, as follows:

Functionality	HL API	IPL API
Enable/disable interrupt monitors	Platform_SetIrqMonitor	Intm_Ip_EnableMonitor / Intm↩ _Ip_DisableMonitor
Ackonwledge a monitored IRQ has been served	Platform_AckIrq	Intm_Ip_AckIrq
Select the monitored IRQ	Platform_SelectMonitoredIrq	Intm_Ip_SelectMonitoredIrq
Set the accepted latency	Platform_SetMonitoredIrqLatency	Intm_Ip_SetLatency
Reset the monitors timer	Platform_ResetIrqMonitorTimer	Intm_Ip_ResetTimer
Retrieve the status (if latency was exceeded)	Platform_GetIrqMonitorStatus	Intm_Ip_GetStatus

3.7 Runtime errors

The driver does not generate DEM errors.

The development errors generated through DET are:

Error Code	Condition triggering the error
PLATFORM_E_PARAM_POINTER	Invalid pointer (null pointer) for parameters passed as reference
PLATFORM_E_PARAM_OUT_OF_RANGE	Parameter out of range
PLATFORM_E_VECTOR_TABLE_READ_ONLY	vector table resides in target flash
PLATFORM_E_PARAM_CONFIG	call from wrong mapped partition

3.8 Symbolic Names Disclaimer

All containers having symbolicNameValue set to TRUE in the AUTOSAR schema will generate defines like:

#define <Mip>Conf_<Container_ShortName>_<Container_ID>

For this reason it is forbidden to duplicate the names of such containers across the RTD configurations 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 driver. All the parameters are described below.

- Module Platform
  - Container GeneralConfiguration
    - Parameter PlatformDevErrorDetect
    - Parameter PlatformIntmConfigurable
    - Parameter PlatformMcmConfigurable
    - Parameter PlatformMpuM7Configurable
    - Parameter PlatformIpAPIsAvailable
    - Parameter PlatformEnableUserModeSupport
    - Parameter PlatformMulticoreSupport
    - Parameter PlatformEnableVtorConfiguration
    - Reference PlatformEcucPartitionRef
  - Container McmConfig
    - Parameter SystemAhbSlavePrio
    - Reference PlatformMcmEcucPartitionRef
    - Container SystemIsrConfig
      - Parameter SystemIsrName
      - Parameter SystemIsrEnabled
  - Container MPU_M7_Configuration
    - Container MPU_M7_ModuleConfig
      - Parameter DefaultMapEnable
      - Parameter RunInHFNMIEnable
      - Parameter MemManageInterruptEnable
      - Parameter PhysicalCoreID
      - Reference Mpu_M7_EcucPartitionRef
      - Container RegionConfig
      - Parameter RegionNumber
      - Parameter StartAddress
      - Parameter EndAddress
      - Parameter RegionSize
      - Parameter MemoryType

      - Parameter AccessRights
      - Parameter OuterCachePolicy
      - Parameter InnerCachePolicy
      - Parameter SubregionMask
      - Parameter Shareable
  - Container IntmConfig
    - Container IntmGenericSettings
      - Parameter IntmEnable
      - Reference PlatformIntmGenericEcucPartitionRef
    - Container IntmChannel
      - Parameter PlatformIntmChannelId
      - Parameter InterruptMonitor
      - Parameter MonitoredIrq
      - Parameter AcceptedLatency
      - Reference PlatformIntmChannelEcucPartitionRef
  - Container IntCtrlConfig
    - Parameter PlatformVtorAddressConfig
    - Reference PlatformNvicEcucPartitionRef
    - Container PlatformIsrConfig
      - Parameter IsrName
      - Parameter IsrEnabled
      - Parameter IsrPriority
      - Parameter IsrHandler
  - Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Platform

Configuration of Platform module.

Included containers:

- GeneralConfiguration

- McmConfig
- MPU_M7_Configuration
- IntmConfig
- IntCtrlConfig
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantSupport	False
supportedConfigVariants	VARIANT-PRE-COMPILE, VARIANT-POST-BUILD

4.2 Container GeneralConfiguration

GeneralConfiguration

This container contains the global configuration parameters of the Non-Autosar Platform driver.

Note: Implementation Specific Parameter.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.3 Parameter PlatformDevErrorDetect

PlatformDevErrorDetect

Switches the Development Error Detection and Notification ON or OFF.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	true

4.4 Parameter PlatformIntmConfigurable

PlatformIntmConfigurable

Check this in order to be able to use the interrupt monitors.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.5 Parameter PlatformMcmConfigurable

PlatformMcmConfigurable

Check this in order to be able to configure Miscellaneous Control settings.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.6 Parameter PlatformMpuM7Configurable

PlatformMpuM7Configurable

Check this in order to be able to use the ARM Cortex M7 MPU.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.7 Parameter PlatformIpAPIsAvailable

PlatformIpAPIsAvailable

Enable or disable IP layer APIs which are not used by APIs at High Level Driver (HLD). Following APIs are affected:

IntCtrl_Ip_SetPending

IntCtrl_Ip_GetPending

IntCtrl_Ip_GetActive

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.8 Parameter PlatformEnableUserModeSupport

When this parameter is enabled, the Platform module will adapt to run from User Mode, with the following measures:

b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode.

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.9 Parameter PlatformMulticoreSupport

This parameter globally enables the possibility to support multicore. If this parmeter is enabled, at least one EcucPartition needs to be defined (in all variants).

NoteThis is an Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.10 Parameter PlatformEnableVtorConfiguration

When this parameter is enabled, the Platform module will allow the user the manually configure the Vector Table Offset Register address:

Note: Implementation Specific Parameter.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.11 Reference PlatformEcucPartitionRef

Maps the Platform driver to zero a multiple ECUC partitions to make the modules API available in this partition.

Note: Each PlatformEcucPartitionRef should map to a M7 core, one by one

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.12 Container McmConfig

Vendor specific:

Miscellaneous Control Configuration

Included subcontainers:

- SystemIsrConfig

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.13 Parameter SystemAhbSlavePrio

Vendor specific:

Configures the access priority on the AHBS port of the Cortex-M7.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	Round_robin
literals	['Round_robin', 'AHB_Slave_priority']

4.14 Reference PlatformMcmEcucPartitionRef

Maps a instance of Mcm to ECUC partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.15 Container SystemIsrConfig

Vendor specific:

Configuration for core-related interrupts.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	7
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.16 Parameter SystemIsrName

Vendor specific:

Interrupt Name.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	FPU_INPUT_DENORMAL_IRQ
literals	['FPU_INPUT_DENORMAL_IRQ', 'FPU_INEXACT_IRQ', 'FPU_UNDE↩ RFLOW_IRQ', 'FPU_OVERFLOW_IRQ', 'FPU_DIVIDE_BY_ZERO_IRQ', 'FPU_INVALID_OPERATION_IRQ', 'TCM_WRITE_ABORT_IRQ']

4.17 Parameter SystemIsrEnabled

Vendor specific: Switch to indicate if the interrupt is enabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.18 Container MPU_M7_Configuration

Configuration for the ARM Cortex M7 MPU.

Included subcontainers:

- MPU_M7_ModuleConfig

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.19 Container MPU_M7_ModuleConfig

Configuration for the ARM Cortex M7 MPU.

Included subcontainers:

- RegionConfig

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.20 Parameter DefaultMapEnable

Vendor specific: Switch to indicate if the default memmory map is enabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.21 Parameter RunInHFNMIEnable

Vendor specific: Switch to indicate if the MPU is enabled in Hard Fault or NMI

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.22 Parameter MemManageInterruptEnable

Vendor specific: Switch to indicate if the Memory Management Interrupt is enabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.23 Parameter PhysicalCoreID

Physical Core ID

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	Cortex_M7_Core0
literals	['Cortex_M7_Core0', 'Cortex_M7_Core1', 'Cortex_M7_Core2']

4.24 Reference Mpu_M7_EcucPartitionRef

Maps an instance of Nvic ECUC partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false

Property	Value
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.25 Container RegionConfig

Vendor specific:

Configuration for interrupt requests.

Warning: This is a precompile configuration.If you uncheck a ISR, you will not be able to enable the

respective channel or error functionality at post build time.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	16
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE

4.26 Parameter RegionNumber

Vendor specific:

Region Number.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	15
min	0

4.27 Parameter StartAddress

Vendor specific:

Start Address.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	4294967294
min	0

4.28 Parameter EndAddress

Vendor specific:

End Address.

Property	Value
type	ECUC-INTEGER-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	4294967295
max	4294967295
min	31

4.29 Parameter RegionSize

Vendor specific:

RegionSize.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	1
max	9223372036854775807
min	-9223372036854775808

4.30 Parameter MemoryType

Vendor specific:

Memory type.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MPU_M7_MEM_STRONG_ORDER
literals	['MPU_M7_MEM_STRONG_ORDER', 'MPU_M7_MEM_DEVICE_SHA↩ RED', 'MPU_M7_MEM_NORMAL_IO_WR_THROUGH', 'MPU_M7_M↩ EM_NORMAL_IO_WR_BACK1', 'MPU_M7_MEM_NORMAL_IO_NO↩ _CACHE', 'MPU_M7_MEM_NORMAL_IO_WR_BACK2', 'MPU_M7_M↩ EM_DEVICE_NOSHARE', 'MPU_M7_MEM_NORMAL_CACHEABLE']

4.31 Parameter AccessRights

Vendor specific:

Access Rights.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MPU_M7_PRIV_UNPRIV_NONE
literals	['MPU_M7_PRIV_UNPRIV_NONE', 'MPU_M7_PRIV_RW_UNPRIV_N↩ ONE', 'MPU_M7_PRIV_RWX_UNPRIV_NONE', 'MPU_M7_PRIV_RW↩ _UNPRIV_R', 'MPU_M7_PRIV_RWX_UNPRIV_RX', 'MPU_M7_PRIV↩ _RW_UNPRIV_RW', 'MPU_M7_PRIV_RWX_UNPRIV_RWX', 'MPU_↩ M7_PRIV_R_UNPRIV_NONE', 'MPU_M7_PRIV_RX_UNPRIV_NONE', 'MPU_M7_PRIV_R_UNPRIV_R', 'MPU_M7_PRIV_RX_UNPRIV_RX']

4.32 Parameter OuterCachePolicy

Vendor specific:

Outer Cache Policy.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MPU_M7_CACHE_POLICY_NO_CACHE
literals	['MPU_M7_CACHE_POLICY_NO_CACHE', 'MPU_M7_CACHE_POLICY_W_BACK_WR_ALLOCATE', 'MPU_M7_CACHE_POLICY_W_THROUGH_NO_W_ALLOCATE', 'MPU_M7_CACHE_POLICY_W_BACK_NO_W_ALLOCATE']

4.33 Parameter InnerCachePolicy

Vendor specific:

Inner Cache Policy.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MPU_M7_CACHE_POLICY_NO_CACHE
literals	['MPU_M7_CACHE_POLICY_NO_CACHE', 'MPU_M7_CACHE_POLICY_W_BACK_WR_ALLOCATE', 'MPU_M7_CACHE_POLICY_W_THROUGH_NO_W_ALLOCATE', 'MPU_M7_CACHE_POLICY_W_BACK_NO_W_ALLOCATE']

4.34 Parameter SubregionMask

Vendor specific:

Sub Region.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	255
min	0

4.35 Parameter Shareable

Vendor specific: Switch to indicate if the region is shareable

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.36 Container IntmConfig

Configuration for interrupt monitors.

Included subcontainers:

- IntmGenericSettings
- IntmChannel

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.37 Container IntmGenericSettings

IntmGenericSettings

This container contains the global configuration parameters interrupt monitor.

Note: Implementation Specific Parameter.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.38 Parameter IntmEnable

Vendor specific:

Enables the interrupt monitoring feature.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.39 Reference PlatformIntmGenericEcucPartitionRef

Maps Intm Generic setting to ECUC partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.40 Container IntmChannel

Vendor specific:

Configuration for interrupt monitors.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	4
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.41 Parameter PlatformIntmChannelId

Channel Id of the interrupt monitor channel.

This value will be assigned to the symbolic name derived of the PlatformIntmChannelId container short name.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	true
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	1
max	3
min	0

4.42 Parameter InterruptMonitor

Vendor specific:

Interrupt monitor index.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	IntMonitor0
literals	['IntMonitor0', 'IntMonitor1', 'IntMonitor2', 'IntMonitor3']

4.43 Parameter MonitoredIrq

Vendor specific:

Monitored interrupt source.

If multicore is support, one interrupt source can be monitored by multiple channels, each channel for each core.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	INT0_IRQn

Property	Value
literals	['INT0_IRQn', 'INT1_IRQn', 'INT2_IRQn', 'INT3_IRQn', 'DMATCD0_IRQn', 'DMATCD1_IRQn', 'DMATCD2_IRQn', 'DMATCD3_IRQn', 'DMATCD4_IRQn', 'DMATCD5_IRQn', 'DMATCD6_IRQn', 'DMATCD7_IRQn', 'DMATCD8_IRQn', 'DMATCD9_IRQn', 'DMATCD10_IRQn', 'DMATCD11_IRQn', 'DMATCD12_IRQn', 'DMATCD13_IRQn', 'DMATCD14_IRQn', 'DMATCD15_IRQn', 'DMATCD16_IRQn', 'DMATCD17_IRQn', 'DMATCD18_IRQn', 'DMATCD19_IRQn', 'DMATCD20_IRQn', 'DMATCD21_IRQn', 'DMATCD22_IRQn', 'DMATCD23_IRQn', 'DMATCD24_IRQn', 'DMATCD25_IRQn', 'DMATCD26_IRQn', 'DMATCD27_IRQn', 'DMATCD28_IRQn', 'DMATCD29_IRQn', 'DMATCD30_IRQn', 'DMATCD31_IRQn', 'ERM_0_IRQn', 'ERM_1_IRQn', 'STM0_IRQn', 'STM1_IRQn', 'STM2_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', 'FLASH_0_IRQn', 'FLASH_1_IRQn', 'FLASH_2_IRQn', 'RGM_IRQn', 'PMC_IRQn', 'SIUL_0_IRQn', 'SIUL_1_IRQn', 'SIUL_2_IRQn', 'SIUL_3_IRQn', 'EMIOS0_0_IRQn', 'EMIOS0_1_IRQn', 'EMIOS0_2_IRQn', 'EMIOS0_3_IRQn', 'EMIOS0_4_IRQn', 'EMIOS0_5_IRQn', 'EMIOS1_0_IRQn', 'EMIOS1_1_IRQn', 'EMIOS1_2_IRQn', 'EMIOS1_3_IRQn', 'EMIOS1_4_IRQn', 'EMIOS1_5_IRQn', 'EMIOS2_0_IRQn', 'EMIOS2_1_IRQn', 'EMIOS2_2_IRQn', 'EMIOS2_3_IRQn', 'EMIOS2_4_IRQn', 'EMIOS2_5_IRQn', 'WKPU_IRQn', 'CMU0_IRQn', 'CMU1_IRQn', 'CMU2_IRQn', 'BCTU_IRQn', 'LCU0_IRQn', 'LCU1_IRQn', 'PIT0_IRQn', 'PIT1_IRQn', 'PIT2_IRQn', 'RTC_IRQn', 'FlexCAN0_0_IRQn', 'FlexCAN0_1_IRQn', 'FlexCAN0_2_IRQn', 'FlexCAN0_3_IRQn', 'FlexCAN1_0_IRQn', 'FlexCAN1_1_IRQn', 'FlexCAN1_2_IRQn', 'FlexCAN2_0_IRQn', 'FlexCAN2_1_IRQn', 'FlexCAN2_2_IRQn', 'FlexCAN3_0_IRQn', 'FlexCAN3_1_IRQn', 'FlexCAN4_0_IRQn', 'FlexCAN4_1_IRQn', 'FlexCAN5_0_IRQn', 'FlexCAN5_1_IRQn', 'FlexCAN6_0_IRQn', 'FlexCAN6_1_IRQn', 'FlexCAN7_0_IRQn', 'FlexCAN7_1_IRQn', 'FlexCAN1_3_IRQn', 'FlexCAN2_3_IRQn', 'FlexCAN3_2_IRQn', 'FlexCAN4_2_IRQn', 'FlexCAN5_2_IRQn', 'FlexCAN6_2_IRQn', 'FlexCAN7_2_IRQn', 'FLEXIO_IRQn', 'LPUART0_IRQn', 'LPUART1_IRQn', 'LPUART2_IRQn', 'LPUART3_IRQn', 'LPUART4_IRQn', 'LPUART5_IRQn', 'LPUART6_IRQn', 'LPUART7_IRQn', 'LPUART8_IRQn', 'LPUART9_IRQn', 'LPUART10_IRQn', 'LPUART11_IRQn', 'LPUART12_IRQn', 'LPUART13_IRQn', 'LPUART14_IRQn', 'LPUART15_IRQn', 'LPI2C0_IRQn', 'LPI2C1_IRQn', 'LPSPI0_IRQn', 'LPSPI1_IRQn', 'LPSPI2_IRQn', 'LPSPI3_IRQn', 'LPSPI4_IRQn', 'LPSPI5_IRQn', 'QSPI_IRQn', 'SAI0_IRQn', 'SAI1_IRQn', 'USDHC_IRQn', 'JDC_IRQn', 'ADC0_IRQn', 'ADC1_IRQn', 'ADC2_IRQn', 'LPCMP0_IRQn', 'LPCMP1_IRQn', 'LPCMP2_IRQn', 'FCCU_0_IRQn', 'FCCU_1_IRQn', 'STCU_LBIST_MBIST_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RX_IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_RX_IRQn', 'HSE_MU1_ORED_IRQn', 'MU_A_0_IRQn', 'MU_A_1_IRQn', 'MU_A_2_IRQn', 'MU_B_0_IRQn', 'MU_B_1_IRQn', 'MU_B_2_IRQn', 'SoC_IRQn', 'SoC2_IRQn', 'XRDC_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_CH0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH1_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_SIC_IRQn', 'GMAC0_SIUC_IRQn']

4.44 Parameter AcceptedLatency

Accepted latency for the monitored interrupt. To disable monitoring when initializing, enter 0 for AcceptedLatency

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	16777213
min	0

4.45 Reference PlatformIntmChannelEcucPartitionRef

Maps Intm channel setting to ECUC partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.46 Container IntCtrlConfig

Configuration for the interrupts.

Included subcontainers:

- PlatformIsrConfig

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.47 Parameter PlatformVtorAddressConfig

Configure the address where the Interrupt Vector starts

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	4294967295
min	0

4.48 Reference PlatformNvicEcucPartitionRef

Maps an instance of Nvic ECUC partitions.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-POST-BUILD: PRE-COMPILE

Property	Value
	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.49 Container PlatformIsrConfig

Vendor specific:

Configuration for interrupt requests.

Warning: This is a precompile configuration.If you uncheck a ISR, you will not be able to enable the

respective channel or error functionality at post build time.

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	0
upperMultiplicity	175
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE

4.50 Parameter IsrName

Vendor specific:

Interrupt Name.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	INT0_IRQn
literals	['INT0_IRQn', 'INT1_IRQn', 'INT2_IRQn', 'INT3_IRQn', 'DMATCD0_IRQn', 'DMATCD1_IRQn', 'DMATCD2_IRQn', 'DMATCD3_IRQn', 'DMATCD4_IRQn', 'DMATCD5_IRQn', 'DMATCD6_IRQn', 'DMATCD7_IRQn', 'DMATCD8_IRQn', 'DMATCD9_IRQn', 'DMATCD10_IRQn', 'DMATCD11_IRQn', 'DMATCD12_IRQn', 'DMATCD13_IRQn', 'DMATCD14_IRQn', 'DMATCD15_IRQn', 'DMATCD16_IRQn', 'DMATCD17_IRQn', 'DMATCD18_IRQn', 'DMATCD19_IRQn', 'DMATCD20_IRQn', 'DMATCD21_IRQn', 'DMATCD22_IRQn', 'DMATCD23_IRQn', 'DMATCD24_IRQn', 'DMATCD25_IRQn', 'DMATCD26_IRQn', 'DMATCD27_IRQn', 'DMATCD28_IRQn', 'DMATCD29_IRQn', 'DMATCD30_IRQn', 'DMATCD31_IRQn', 'ERM_0_IRQn', 'ERM_1_IRQn', 'MCM_IRQn', 'STM0_IRQn', 'STM1_IRQn', 'STM2_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', 'CTI0_IRQn', 'CTI1_IRQn', 'FLASH_0_IRQn', 'FLASH_1_IRQn', 'FLASH_2_IRQn', 'RGM_IRQn', 'PMC_IRQn', 'SIUL_0_IRQn', 'SIUL_1_IRQn', 'SIUL_2_IRQn', 'SIUL_3_IRQn', 'EMIOS0_0_IRQn', 'EMIOS0_1_IRQn', 'EMIOS0_2_IRQn', 'EMIOS0_3_IRQn', 'EMIOS0_4_IRQn', 'EMIOS0_5_IRQn', 'EMIOS1_0_IRQn', 'EMIOS1_1_IRQn', 'EMIOS1_2_IRQn', 'EMIOS1_3_IRQn', 'EMIOS1_4_IRQn', 'EMIOS1_5_IRQn', 'EMIOS2_0_IRQn', 'EMIOS2_1_IRQn', 'EMIOS2_2_IRQn', 'EMIOS2_3_IRQn', 'EMIOS2_4_IRQn', 'EMIOS2_5_IRQn', 'WKPU_IRQn', 'CMU0_IRQn', 'CMU1_IRQn', 'CMU2_IRQn', 'BCTU_IRQn', 'LCU0_IRQn', 'LCU1_IRQn', 'PIT0_IRQn', 'PIT1_IRQn', 'PIT2_IRQn', 'RTC_IRQn', 'FlexCAN0_0_IRQn', 'FlexCAN0_1_IRQn', 'FlexCAN0_2_IRQn', 'FlexCAN0_3_IRQn', 'FlexCAN1_0_IRQn', 'FlexCAN1_1_IRQn', 'FlexCAN1_2_IRQn', 'FlexCAN2_0_IRQn', 'FlexCAN2_1_IRQn', 'FlexCAN2_2_IRQn', 'FlexCAN3_0_IRQn', 'FlexCAN3_1_IRQn', 'FlexCAN4_0_IRQn', 'FlexCAN4_1_IRQn', 'FlexCAN5_0_IRQn', 'FlexCAN5_1_IRQn', 'FlexCAN6_0_IRQn', 'FlexCAN6_1_IRQn', 'FlexCAN7_0_IRQn', 'FlexCAN7_1_IRQn', 'FlexCAN1_3_IRQn', 'FlexCAN2_3_IRQn', 'FlexCAN3_2_IRQn', 'FlexCAN4_2_IRQn', 'FlexCAN5_2_IRQn', 'FlexCAN6_2_IRQn', 'FlexCAN7_2_IRQn', 'FLEXIO_IRQn', 'LPUART0_IRQn', 'LPUART1_IRQn', 'LPUART2_IRQn', 'LPUART3_IRQn', 'LPUART4_IRQn', 'LPUART5_IRQn', 'LPUART6_IRQn', 'LPUART7_IRQn', 'LPUART8_IRQn', 'LPUART9_IRQn', 'LPUART10_IRQn', 'LPUART11_IRQn', 'LPUART12_IRQn', 'LPUART13_IRQn', 'LPUART14_IRQn', 'LPUART15_IRQn', 'LPI2C0_IRQn', 'LPI2C1_IRQn', 'LPSPI0_IRQn', 'LPSPI1_IRQn', 'LPSPI2_IRQn', 'LPSPI3_IRQn', 'LPSPI4_IRQn', 'LPSPI5_IRQn', 'QSPI_IRQn', 'SAI0_IRQn', 'SAI1_IRQn', 'USDHC_IRQn', 'JDC_IRQn', 'ADC0_IRQn', 'ADC1_IRQn', 'ADC2_IRQn', 'LPCMP0_IRQn', 'LPCMP1_IRQn', 'LPCMP2_IRQn', 'FCCU_0_IRQn', 'FCCU_1_IRQn', 'STCU_LBIST_MBIST_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RX_IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_RX_IRQn', 'HSE_MU1_ORED_IRQn', 'MU_A_0_IRQn', 'MU_A_1_IRQn', 'MU_A_2_IRQn', 'MU_B_0_IRQn', 'MU_B_1_IRQn', 'MU_B_2_IRQn', 'SoC_IRQn', 'SoC2_IRQn', 'XRDC_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_CH0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH1_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_SIC_IRQn', 'GMAC0_SIUC_IRQn']

4.51 Parameter IsrEnabled

Vendor specific: Switch to indicate if the interrupt is enabled

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.52 Parameter IsrPriority

Priority of the interrupt interrupt

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	False
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	0
max	255
min	0

4.53 Parameter IsrHandler

Function to be installed as the interrupt handler.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	undefined_handler

4.54 Container CommonPublishedInformation

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

Included subcontainers:

- None

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.55 Parameter ArReleaseMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	4
max	4
min	4

4.56 Parameter ArReleaseMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	7
max	7
min	7

4.57 Parameter ArReleaseRevisionVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	0
min	0

4.58 Parameter ModuleId

Module ID of this module from Module List.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	255
max	255
min	255

4.59 Parameter SwMajorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE

Property	Value
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	3
max	3
min	3

4.60 Parameter SwMinorVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	0
min	0

4.61 Parameter SwPatchVersion

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0

Property	Value
max	0
min	0

4.62 Parameter VendorApiInfix

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>.

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.

Property	Value
type	ECUC-STRING-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	

4.63 Parameter VendorId

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	43
max	43
min	43

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

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 Interrupt Controller IP

6.1.1 Detailed Description

Data Structures

- struct IntCtrl_Ip_IrqConfigType

  Structure storing the state and priority configuration for an interrupt request. More...
- struct IntCtrl_Ip_CtrlConfigType

  Structure storing the list of state configurations for all configured interrupts. More...

Types Reference

- typedef void(∗ IntCtrl_Ip_IrqHandlerType) (void)

  Interrupt handler type.

Enum Reference

- enum IntCtrl_Ip_StatusType

  Enumeration listing the possible error codes returned by IntCtrl_Ip API.
- enum IntCtrl_Ip_IrqTargetType

  Enumeration listing the available target cores for an inter-core interrupt.

Function Reference

- IntCtrl_Ip_StatusType IntCtrl_Ip_Init (const IntCtrl_Ip_CtrlConfigType *pIntCtrlCtrlConfig)

  Initializes the configured interrupts at interrupt controller level.
- void IntCtrl_Ip_InstallHandler (IRQn_Type eIrqNumber, const IntCtrl_Ip_IrqHandlerType pfNewHandler, IntCtrl_Ip_IrqHandlerType *const pfOldHandler)

  Installs a handler for an IRQ.
- void IntCtrl_Ip_EnableIrq (IRQn_Type eIrqNumber)

  Enables an interrupt request.
- void IntCtrl_Ip_DisableIrq (IRQn_Type eIrqNumber)

  Disables an interrupt request.
- void IntCtrl_Ip_SetPriority (IRQn_Type eIrqNumber, uint8 u8Priority)

  Sets the priority for an interrupt request.
- uint8 IntCtrl_Ip_GetPriority (IRQn_Type eIrqNumber)

  Gets the priority for an interrupt request.
- void IntCtrl_Ip_ClearPending (IRQn_Type eIrqNumber)

  Clears the pending flag for an interrupt request.

6.1.2 Data Structure Documentation

6.1.2.1 struct IntCtrl_Ip_IrqConfigType

Structure storing the state and priority configuration for an interrupt request.

Definition at line 114 of file IntCtrl_Ip_TypesDef.h.

Data Fields

- IRQn_Type eIrqNumber

  Interrupt number.
- boolean bIrqEnabled

  Interrupt state (enabled/disabled)
- uint8 u8IrqPriority

  Interrupt priority.
- IntCtrl_Ip_IrqHandlerType pfHandler

  Interrupt handler.

6.1.2.1.1 Field Documentation

6.1.2.1.1.1 eIrqNumber `IRQn_Type eIrqNumber`

Interrupt number.

Definition at line 117 of file IntCtrl_Ip_TypesDef.h.

6.1.2.1.1.2 bIrqEnabled `boolean bIrqEnabled`

Interrupt state (enabled/disabled)

Definition at line 119 of file IntCtrl_Ip_TypesDef.h.

6.1.2.1.1.3 u8IrqPriority `uint8 u8IrqPriority`

Interrupt priority.

Definition at line 121 of file IntCtrl_Ip_TypesDef.h.

6.1.2.1.1.4 pfHandler `IntCtrl_Ip_IrqHandlerType pfHandler`

Interrupt handler.

Definition at line 123 of file IntCtrl_Ip_TypesDef.h.

6.1.2.2 struct IntCtrl_Ip_CtrlConfigType

Structure storing the list of state configurations for all configured interrupts.

Definition at line 130 of file IntCtrl_Ip_TypesDef.h.

Data Fields

- uint32 u32ConfigIrqCount

  Number of configured interrupts.
- const IntCtrl_Ip_IrqConfigType ∗ aIrqConfig

  List of interrupts configurations.

6.1.2.2.1 Field Documentation

6.1.2.2.1.1 u32ConfigIrqCount `uint32 u32ConfigIrqCount`

Number of configured interrupts.

Definition at line 133 of file IntCtrl_Ip_TypesDef.h.

6.1.2.2.1.2 aIrqConfig `const IntCtrl_Ip_IrqConfigType* aIrqConfig`

List of interrupts configurations.

Definition at line 139 of file IntCtrl_Ip_TypesDef.h.

6.1.3 Types Reference

6.1.3.1 IntCtrl_Ip_IrqHandlerType

```
typedef void(* IntCtrl_Ip_IrqHandlerType) (void)
```

Interrupt handler type.

Definition at line 78 of file IntCtrl_Ip_TypesDef.h.

6.1.4 Enum Reference

6.1.4.1 IntCtrl_Ip_StatusType

```
enum IntCtrl_Ip_StatusType
```

Enumeration listing the possible error codes returned by IntCtrl_Ip API.

Enumerator

INTCTRL_IP_STATUS_SUCCESS	Status SUCCESS.
INTCTRL_IP_STATUS_ERROR	Status ERROR.

Definition at line 146 of file IntCtrl_Ip_TypesDef.h.

6.1.4.2 IntCtrl_Ip_IrqTargetType

```
enum IntCtrl_Ip_IrqTargetType
```

Enumeration listing the available target cores for an inter-core interrupt.

Enumerator

INTCTRL_IP_TARGET_SELF	Interrupt request targeted to the same core that triggers it.
INTCTRL_IP_TARGET_OTHERS	Interrupt request targeted to all the other cores.
INTCTRL_IP_TARGET_CP0	Interrupt request targeted to core 0.
INTCTRL_IP_TARGET_CP1	Interrupt request targeted to core 1.
INTCTRL_IP_TARGET_CP2	Interrupt request targeted to core 2.

Definition at line 160 of file IntCtrl_Ip_TypesDef.h.

6.1.5 Function Reference

6.1.5.1 IntCtrl_Ip_Init()

```
IntCtrl_Ip_StatusType IntCtrl_Ip_Init (
            const IntCtrl_Ip_CtrlConfigType * pIntCtrlCtrlConfig )
```

Initializes the configured interrupts at interrupt controller level.

This function is non-reentrant and initializes the interrupts.

Parameters

in	*pIntCtrlCtrlConfig*	pointer to configuration structure for interrupts.

Returns

IntCtrl_Ip_StatusType: error code.

6.1.5.2 IntCtrl_Ip_InstallHandler()

```
void IntCtrl_Ip_InstallHandler (
            IRQn_Type eIrqNumber,
            const IntCtrl_Ip_IrqHandlerType pfNewHandler,
            IntCtrl_Ip_IrqHandlerType *const pfOldHandler )
```

Installs a handler for an IRQ.

This function is non-reentrant; it installs an new ISR for an interrupt line.

Note

This function works only when the interrupt vector table resides in RAM.

Parameters

in	*eIrqNumber*	interrupt number.
in	*pfNewHandler*	function pointer for the new handler.
out	*pfOldHandler*	stores the address of the old interrupt handler.

Returns

void.

6.1.5.3 IntCtrl_Ip_EnableIrq()

```
void IntCtrl_Ip_EnableIrq (
            IRQn_Type eIrqNumber )
```

Enables an interrupt request.

This function is non-reentrant; it enables the interrupt request at interrupt controller level.

Parameters

in	*eIrqNumber*	interrupt number to be enabled.

Returns

void.

6.1.5.4 IntCtrl_Ip_DisableIrq()

```
void IntCtrl_Ip_DisableIrq (
            IRQn_Type eIrqNumber )
```

Disables an interrupt request.

This function is non-reentrant; it disables the interrupt request at interrupt controller level.

Parameters

in	*eIrqNumber*	interrupt number to be disabled.

Returns

void.

6.1.5.5 IntCtrl_Ip_SetPriority()

```
void IntCtrl_Ip_SetPriority (
            IRQn_Type eIrqNumber,
            uint8 u8Priority )
```

Sets the priority for an interrupt request.

This function is non-reentrant; it sets the priority for the interrupt request.

Parameters

in	*eIrqNumber*	interrupt number for which the priority is set.
in	*u8Priority*	the priority to be set.

Returns

void.

6.1.5.6 IntCtrl_Ip_GetPriority()

```
uint8 IntCtrl_Ip_GetPriority (
            IRQn_Type eIrqNumber )
```

Gets the priority for an interrupt request.

This function is non-reentrant; it retrieves the priority for the interrupt request.

Parameters

in	*eIrqNumber*	interrupt number for which the priority is set.

Returns

uint8: the priority of the interrupt.

6.1.5.7 IntCtrl_Ip_ClearPending()

```
void IntCtrl_Ip_ClearPending (
            IRQn_Type eIrqNumber )
```

Clears the pending flag for an interrupt request.

This function is reentrant; it clears the pending flag for the interrupt request.

Parameters

in	*eIrqNumber*	interrupt number for which the pending flag is cleared.

Returns

void.

6.2 Interrupt Monitor IP

6.2.1 Detailed Description

6.3 Mpu M7 IPV Driver

6.3.1 Detailed Description

6.4 Platform

6.4.1 Detailed Description

Modules

- Interrupt Controller IP
- Interrupt Monitor IP
- System IP

Data Structures

- struct Platform_ConfigType

  Configuration structure for PLATFORM CDD. More...

Macros

- #define PLATFORM_E_PARAM_POINTER

  All API's having pointers as parameters shall return this error if called with with a NULL value.
- #define PLATFORM_E_PARAM_OUT_OF_RANGE

  Error returned for parameters out of range.
- #define PLATFORM_E_PARAM_CONFIG

  If DET error reporting is enabled, the PLATFORM will check upon each API call if the requested resource is configured to be available on the current core, and in case of error will return PLATFORM_E_PARAM_CONFIG.
- #define PLATFORM_E_PARAM_CHANNEL

  API service called with wrong parameter of Channel.
- #define PLATFORM_INIT_ID

  Service ID of Platform_Init function.
- #define PLATFORM_SET_IRQ_ID

  Service ID of Platform_SetIrq function.
- #define PLATFORM_SET_IRQ_PRIO_ID

  Service ID of Platform_SetIrqPriority function.
- #define PLATFORM_GET_IRQ_PRIO_ID

  Service ID of Platform_GetIrqPriority function.
- #define PLATFORM_INSTALL_HANDLER_ID

  Service ID of Platform_InstallIrqHandler function.
- #define PLATFORM_SET_IRQ_MONITOR_ID

  Service ID of Platform_SetIrqMonitor function.
- #define PLATFORM_ACK_IRQ_ID

  Service ID of Platform_AckIrq function.
- #define PLATFORM_SELECT_MONITORED_IRQ_ID

  Service ID of Platform_SelectMonitoredIrq function.
- #define PLATFORM_SET_MONITORED_IRQ_LATENCY_ID

  Service ID of Platform_SetMonitoredIrqLatency function.
- #define PLATFORM_RESET_IRQ_MONITOR_TIMER_ID

  Service ID of Platform_ResetIrqMonitorTimer function.
- #define PLATFORM_GET_IRQ_MONITOR_STATUS_ID

  Service ID of Platform_GetIrqMonitorStatus function.
- #define PLATFORM_MRUTRANSMIT_ID

  Service ID of Platform_MruTransmit function.

Types Reference

- typedef IntCtrl_Ip_IrqHandlerType Platform_IrqHandlerType

  Interrupt handler type definition for PLATFORM CDD.

Function Reference

- void Platform_Init (const Platform_ConfigType ∗pConfig)

  Initializes the paltform settings based on user configuration.
- Std_ReturnType Platform_SetIrq (IRQn_Type eIrqNumber, boolean bEnable)

  Configures (enables/disables) an interrupt request.
- Std_ReturnType Platform_SetIrqPriority (IRQn_Type eIrqNumber, uint8 u8Priority)

  Configures the priority of an interrupt request.
- Std_ReturnType Platform_GetIrqPriority (IRQn_Type eIrqNumber, uint8 ∗u8Priority)

  Returns the priority of an interrupt request.
- Std_ReturnType Platform_InstallIrqHandler (IRQn_Type eIrqNumber, const Platform_IrqHandlerType pfNewHandler, Platform_IrqHandlerType ∗const pfOldHandler)

  Installs a new handler for an interrupt request.
- void Platform_SetIrqMonitor (boolean bEnable)

  Configures (enables/disables) the interrupt monitor.
- void Platform_AckIrq (IRQn_Type eIrqNumber)

  Acknowledges a monitored interrupt has been served.
- void Platform_SelectMonitoredIrq (uint8 u8Channel, IRQn_Type eIrqNumber)

  Selects an interrupt to monitor.
- void Platform_SetMonitoredIrqLatency (uint8 u8Channel, uint32 u32Latency)

  Sets the latency for a monitored interrupt.
- void Platform_ResetIrqMonitorTimer (uint8 u8Channel)

  Resets the timer for an interrupt monitor.
- Std_ReturnType Platform_GetIrqMonitorStatus (uint8 u8Channel, boolean ∗bLatencyExceeded)

  Retrieves the status of an interrupt monitor.

6.4.2 Data Structure Documentation

6.4.2.1 struct Platform_ConfigType

Configuration structure for PLATFORM CDD.

Definition at line 188 of file Platform_TypesDef.h.

Data Fields

- const Platform_Ipw_ConfigType ∗ pIpwConfig

  Reference to IPW structure.

6.4.2.1.1 Field Documentation

6.4.2.1.1.1 pIpwConfig `const Platform_Ipw_ConfigType* pIpwConfig`

Reference to IPW structure.

Definition at line 191 of file Platform_TypesDef.h.

6.4.3 Macro Definition Documentation

6.4.3.1 PLATFORM_E_PARAM_POINTER

```
#define PLATFORM_E_PARAM_POINTER
```

All API's having pointers as parameters shall return this error if called with with a NULL value.

Definition at line 85 of file Platform_TypesDef.h.

6.4.3.2 PLATFORM_E_PARAM_OUT_OF_RANGE

```
#define PLATFORM_E_PARAM_OUT_OF_RANGE
```

Error returned for parameters out of range.

Definition at line 91 of file Platform_TypesDef.h.

6.4.3.3 PLATFORM_E_PARAM_CONFIG

```
#define PLATFORM_E_PARAM_CONFIG
```

If DET error reporting is enabled, the PLATFORM will check upon each API call if the requested resource is configured to be available on the current core, and in case of error will return PLATFORM_E_PARAM_CONFIG.

Definition at line 100 of file Platform_TypesDef.h.

6.4.3.4 PLATFORM_E_PARAM_CHANNEL

```
#define PLATFORM_E_PARAM_CHANNEL
```

API service called with wrong parameter of Channel.

Definition at line 106 of file Platform_TypesDef.h.

6.4.3.5 PLATFORM_INIT_ID

```
#define PLATFORM_INIT_ID
```

Service ID of Platform_Init function.

Parameter used when raising an error/exception

Definition at line 112 of file Platform_TypesDef.h.

6.4.3.6 PLATFORM_SET_IRQ_ID

```
#define PLATFORM_SET_IRQ_ID
```

Service ID of Platform_SetIrq function.

Parameter used when raising an error/exception

Definition at line 118 of file Platform_TypesDef.h.

6.4.3.7 PLATFORM_SET_IRQ_PRIO_ID

```
#define PLATFORM_SET_IRQ_PRIO_ID
```

Service ID of Platform_SetIrqPriority function.

Parameter used when raising an error/exception

Definition at line 124 of file Platform_TypesDef.h.

6.4.3.8 PLATFORM_GET_IRQ_PRIO_ID

```
#define PLATFORM_GET_IRQ_PRIO_ID
```

Service ID of Platform_GetIrqPriority function.

Parameter used when raising an error/exception

Definition at line 130 of file Platform_TypesDef.h.

6.4.3.9 PLATFORM_INSTALL_HANDLER_ID

```
#define PLATFORM_INSTALL_HANDLER_ID
```

Service ID of Platform_InstallIrqHandler function.

Parameter used when raising an error/exception

Definition at line 136 of file Platform_TypesDef.h.

6.4.3.10 PLATFORM_SET_IRQ_MONITOR_ID

```
#define PLATFORM_SET_IRQ_MONITOR_ID
```

Service ID of Platform_SetIrqMonitor function.

Parameter used when raising an error/exception

Definition at line 142 of file Platform_TypesDef.h.

6.4.3.11 PLATFORM_ACK_IRQ_ID

```
#define PLATFORM_ACK_IRQ_ID
```

Service ID of Platform_AckIrq function.

Parameter used when raising an error/exception

Definition at line 148 of file Platform_TypesDef.h.

6.4.3.12 PLATFORM_SELECT_MONITORED_IRQ_ID

```
#define PLATFORM_SELECT_MONITORED_IRQ_ID
```

Service ID of Platform_SelectMonitoredIrq function.

Parameter used when raising an error/exception

Definition at line 154 of file Platform_TypesDef.h.

6.4.3.13 PLATFORM_SET_MONITORED_IRQ_LATENCY_ID

```
#define PLATFORM_SET_MONITORED_IRQ_LATENCY_ID
```

Service ID of Platform_SetMonitoredIrqLatency function.

Parameter used when raising an error/exception

Definition at line 160 of file Platform_TypesDef.h.

6.4.3.14 PLATFORM_RESET_IRQ_MONITOR_TIMER_ID

```
#define PLATFORM_RESET_IRQ_MONITOR_TIMER_ID
```

Service ID of Platform_ResetIrqMonitorTimer function.

Parameter used when raising an error/exception

Definition at line 166 of file Platform_TypesDef.h.

6.4.3.15 PLATFORM_GET_IRQ_MONITOR_STATUS_ID

```
#define PLATFORM_GET_IRQ_MONITOR_STATUS_ID
```

Service ID of Platform_GetIrqMonitorStatus function.

Parameter used when raising an error/exception

Definition at line 172 of file Platform_TypesDef.h.

6.4.3.16 PLATFORM_MRUTRANSMIT_ID

```
#define PLATFORM_MRUTRANSMIT_ID
```

Service ID of Platform_MruTransmit function.

Parameter used when raising an error/exception

Definition at line 178 of file Platform_TypesDef.h.

6.4.4 Types Reference

6.4.4.1 Platform_IrqHandlerType

```
typedef IntCtrl_Ip_IrqHandlerType Platform_IrqHandlerType
```

Interrupt handler type definition for PLATFORM CDD.

Definition at line 202 of file Platform_TypesDef.h.

6.4.5 Function Reference

6.4.5.1 Platform_Init()

```
void Platform_Init (
            const Platform_ConfigType * pConfig )
```

Initializes the paltform settings based on user configuration.

This function is non-reentrant; it initializes the interrupts, interrupt monitors (if available), as well as other platform specific settings as defined for each SoC.

Parameters

in	*pConfig*	pointer to platform configuration structure.

Returns

void

6.4.5.2 Platform_SetIrq()

```
Std_ReturnType Platform_SetIrq (
            IRQn_Type eIrqNumber,
            boolean bEnable )
```

Configures (enables/disables) an interrupt request.

This function is non-reentrant; it enables/disables the selected interrupt.

Parameters

in	*eIrqNumber*	interrupt to be configured.
in	*bEnable*	TRUE - enable interrupt, FALSE - disable interrupt.

Returns

Std_ReturnType: E_OK/E_NOT_OK; specific errors are reported through DET.

6.4.5.3 Platform_SetIrqPriority()

```
Std_ReturnType Platform_SetIrqPriority (
            IRQn_Type eIrqNumber,
            uint8 u8Priority )
```

Configures the priority of an interrupt request.

This function is non-reentrant; it sets the priority for the selected interrupt.

Parameters

in	*eIrqNumber*	interrupt number for which priority is configured.
in	*u8Priority*	desired priority of the interrupt.

Returns

Std_ReturnType: E_OK/E_NOT_OK; specific errors are reported through DET.

6.4.5.4 Platform_GetIrqPriority()

```
Std_ReturnType Platform_GetIrqPriority (
            IRQn_Type eIrqNumber,
            uint8 * u8Priority )
```

Returns the priority of an interrupt request.

This function is non-reentrant; it retrieves the current priority of the selected interrupt.

Parameters

in	*eIrqNumber*	interrupt number for which priority is returned.
out	*u8Priority*	output parameter storing the priority of the interrupt.

Returns

Std_ReturnType: E_OK/E_NOT_OK; specific errors are reported through DET.

6.4.5.5 Platform_InstallIrqHandler()

```
Std_ReturnType Platform_InstallIrqHandler (
            IRQn_Type eIrqNumber,
            const Platform_IrqHandlerType pfNewHandler,
            Platform_IrqHandlerType *const pfOldHandler )
```

Installs a new handler for an interrupt request.

This function is non-reentrant; it replaces the current interrupt handler for the selected interrupt with the new function provided as the second parameter. The address of the old handler can be optionally stored in the third parameter.

Parameters

in	*eIrqNumber*	interrupt number for which priority is returned.
in	*pfNewHandler*	function pointer for the new handler.
out	*pfOldHandler*	function pointer that will store the address of the old handler

Note

- this parameter can be passed as NULL if not needed.

Returns

pfOldHandler: E_OK/E_NOT_OK; specific errors are reported through DET.

6.4.5.6 Platform_SetIrqMonitor()

```
void Platform_SetIrqMonitor (
            boolean bEnable )
```

Configures (enables/disables) the interrupt monitor.

This function is reentrant; it enables/disables the interrupt monitoring feature.

Parameters

in	*bEnable*	TRUE - enable interrupt monitoring, FALSE - disable interrupt monitoring.

Returns

void

6.4.5.7 Platform_AckIrq()

```
void Platform_AckIrq (
            IRQn_Type eIrqNumber )
```

Acknowledges a monitored interrupt has been served.

This function is reentrant; it confirms that a monitored interrupt has been served.

Parameters

in	*bEnable*	enable or disable monitoring, TRUE-enable FALSE-disablw.

Returns

void.

6.4.5.8 Platform_SelectMonitoredIrq()

```
void Platform_SelectMonitoredIrq (
            uint8 u8Channel,
            IRQn_Type eIrqNumber )
```

Selects an interrupt to monitor.

This function is reentrant; it selects which interrupt is monitored on which interrupt monitor.

Parameters

in	*u8Channel*	Logical channel of interrupt monitor used.
in	*eIrqNumber*	the interrupt vector to be monitored.

Returns

void.

6.4.5.9 Platform_SetMonitoredIrqLatency()

```
void Platform_SetMonitoredIrqLatency (
            uint8 u8Channel,
            uint32 u32Latency )
```

Sets the latency for a monitored interrupt.

This function is reentrant; it sets the accepted latency for the monitored interrupt.

Parameters

in	*u8Channel*	Logical channel of interrupt monitor used.
in	*u32Latency*	The accepted latency for the monitored interrupt.

Returns

void.

6.4.5.10 Platform_ResetIrqMonitorTimer()

```
void Platform_ResetIrqMonitorTimer (
            uint8 u8Channel )
```

Resets the timer for an interrupt monitor.

This function is reentrant; it resets the timer for the interrupt monitor.

Parameters

in	*u8Channel*	Logical channel of interrupt monitor used.

Returns

void.

6.4.5.11 Platform_GetIrqMonitorStatus()

```
Std_ReturnType Platform_GetIrqMonitorStatus (
            uint8 u8Channel,
            boolean * bLatencyExceeded )
```

Retrieves the status of an interrupt monitor.

This function shall return whether the accepted latency for the monitored IRQs ∗ has been exceeded.

Parameters

in	*u8Channel*	: Logical channel of interrupt monitor used.
out	*bLatencyExceeded*	: output status parameter: TRUE - latency exceeded, FALSE ∗ latency not exceeded

Returns

Std_ReturnType: E_OK/E_NOT_OK; specific errors are reported through DET.

6.5 System IP

6.5.1 Detailed Description

Function Reference

- void System_Ip_SetAhbSlavePriority (boolean bPriority)

  Selects the access priority on the AHBS port of the Cortex-M7.
- void System_Ip_ConfigIrq (System_Ip_IrqType eIrq, boolean bEnable)

  Enables/disables core-related interrupt exceptions.
- void System_Ip_ClearWriteAbortFlag (void)

  Clears Write Abort on Slave flag.
- uint32 System_Ip_GetPlatformRevision (void)

  Returns platform revision.

6.5.2 Function Reference

6.5.2.1 System_Ip_SetAhbSlavePriority()

```
void System_Ip_SetAhbSlavePriority (
            boolean bPriority )
```

Selects the access priority on the AHBS port of the Cortex-M7.

This function is non-reentrant and configures the AHB slave priority.

Parameters

in	*bPriority*	FALSE - round-robin arbitration scheme, TRUE - AHB-slave access has priority over a core access

Returns

void

6.5.2.2 System_Ip_ConfigIrq()

```
void System_Ip_ConfigIrq (
            System_Ip_IrqType eIrq,
            boolean bEnable )
```

Enables/disables core-related interrupt exceptions.

This function is non-reentrant and configures core-related interrupt exceptions, as defined per each platform.

Parameters

in	*eIrq*	core-related interrupt event.
in	*bEnable*	FALSE - disable interrupt, TRUE - enable interrupt.

Returns

void

6.5.2.3 System_Ip_ClearWriteAbortFlag()

```
void System_Ip_ClearWriteAbortFlag (
            void  )
```

Clears Write Abort on Slave flag.

This function is reentrant and clears the flag indicating when a write abort has occurred on the AHBS interface.

Returns

void

6.5.2.4 System_Ip_GetPlatformRevision()

```
uint32 System_Ip_GetPlatformRevision (
            void  )
```

Returns platform revision.

This function is reentrant and returns a software-visible revision number.

Returns

void

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support