User Manual

for S32 PLATFORM Driver

Document Number: UM11PLATFORMASR4.4 Rev0000R4.0.0 Rev. 1.0

1	Revision History	2
2	Introduction	3
	2.1 Supported Derivatives	3
	2.2 Overview	4
	2.3 About This Manual	4
	2.4 Acronyms and Definitions	5
	2.5 Reference List	5
3	Driver	7
	3.1 Requirements	7
	3.2 Driver Design Summary	7
	3.3 Hardware Resources	9
	3.4 Deviations from Requirements	9
	3.5 Driver Limitations	9
	3.6 Driver usage and configuration tips	9
	3.6.1 Initialization	10
	3.6.2 Handling the interrupts	10
	3.7 Runtime errors	10
	3.8 Symbolic Names Disclaimer	10
4	Tresos Configuration Plug-in	11
	4.1 Module Platform	12
	4.2 Container GeneralConfiguration	12
	4.3 Parameter PlatformDevErrorDetect	
	4.4 Parameter PlatformMcmConfigurable	13
	4.5 Parameter PlatformMscmConfigurable	14
	4.6 Parameter PlatformIpAPIsAvailable	14
	4.7 Parameter PlatformEnableUserModeSupport	15
	4.8 Parameter PlatformMulticoreSupport	16
	4.9 Parameter PlatformEnableVtorConfiguration	16
	4.10 Reference PlatformEcucPartitionRef	17
	4.11 Container McmConfig	17
	4.12 Parameter SystemAhbSlavePrio	17
	4.13 Reference PlatformMcmEcucPartitionRef	18
	4.14 Container SystemIsrConfig	18
	4.15 Parameter SystemIsrName	
	4.16 Parameter SystemIsrEnabled	
	4.17 Container IntCtrlConfig	
	4.18 Parameter PlatformVtorAddressConfig	
	4.19 Reference PlatformNvicEcucPartitionRef	

4.20 Container PlatformIsrConfig	 . 21
4.21 Parameter IsrName	 . 22
4.22 Parameter IsrEnabled	 . 24
4.23 Parameter IsrPriority	 . 24
4.24 Container MscmConfig	 . 25
4.25 Reference PlatformGenericInterruptEcucPartitionRef	 . 25
4.26 Container PlatformIsrConfig	 . 25
4.27 Parameter IsrName	 . 26
4.28 Parameter IsrTargetCore0	 . 28
4.29 Parameter IsrTargetCore1	 . 28
4.30 Parameter IsrTargetCore2	 . 29
4.31 Parameter IsrTargetCore3	 . 29
4.32 Parameter IsrTargetCore4	 . 30
4.33 Parameter IsrHandler	 . 30
4.34 Container CommonPublishedInformation	 . 30
4.35 Parameter ArReleaseMajorVersion	 . 31
4.36 Parameter ArReleaseMinorVersion	 . 31
4.37 Parameter ArReleaseRevisionVersion	 . 32
4.38 Parameter ModuleId	 . 32
4.39 Parameter SwMajorVersion	 . 33
4.40 Parameter SwMinorVersion	 . 33
4.41 Parameter SwPatchVersion	 . 34
4.42 Parameter VendorApiInfix	 . 34
4.43 Parameter VendorId	 . 35
Module Index	36
5.1 Software Specification	
oli bolonale specimentoli i i i i i i i i i i i i i i i i i i	 . 00
Module Documentation	37
6.1 Interrupt Controller IP	 . 37
6.1.1 Detailed Description	 . 37
6.1.2 Data Structure Documentation	 . 38
6.1.3 Types Reference	 . 41
6.1.4 Enum Reference	 . 41
6.1.5 Function Reference	 . 42
6.2 Platform	 . 48
6.2.1 Detailed Description	 . 48
6.2.2 Data Structure Documentation	 . 49
6.2.3 Macro Definition Documentation	 . 50
6.2.4 Types Reference	 . 53
6.2.5 Function Reference	 . 53

6.3 System IP	57
6.3.1 Detailed Description	57
6.3.2 Function Reference	57

Chapter 1

Revision History

Revision Date Author		Author	Description
1.0	31.10.2022	NXP RTD Team	Prepared for release S32 RTD AUTOSAR 4.4 Version 4.0.0 Release

Chapter 2

Introduction

- Supported Derivatives
- Overview
- About This Manual
- Acronyms and Definitions
- Reference List

This User Manual describes the NXP Semiconductor PLATFORM driver for S32. 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:

- s32g274a bga525
- s32g254a_bga525
- s32g233a_bga525
- s32g234m_bga525
- $s32g378a_bga525$
- s32g379a_bga525
- s32g398a_bga525
- s32g399a_bga525
- s32g338m_bga525
- $s32g339m_bga525$
- s32g358a_bga525
- $s32g359a_bga525$
- s32r45_bga780

All of the above microcontroller devices are collectively named as S32.

Introduction

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 4.4.0
2	Specification of Communication Stack Types	AUTOSAR Release 4.4.0
3	Specification of Compiler Abstraction	AUTOSAR Release 4.4.0
4	Specification of Platform Types	AUTOSAR Release 4.4.0
5	Specification of Standard Types	AUTOSAR Release 4.4.0
6	S32G2 Reference Manual	Rev. 5, May 2022
7	S32G3 Reference Manual	Rev. 2 Draft C, June 2022
8	S32R45 Reference Manual	Rev. 3, 12/2021
9	S32G2 Data Sheet	Rev. 5, May 2022
10	S32G3 Data Sheet	Rev. 2 Draft B, June 2022
11	S32R45 Data Sheet	Rev. 2, 12/2021
12	VR5510 Data Sheet	Rev. 5, April 2022
13	S32G2 Errata Document	Mask Set Errata for Mask 0P77B, Rev. 2.4

Introduction

#	Title	Version
14	S32G3 Errata Document	Mask Set Errata for Mask 0P72B, Rev. 1.1
15	S32R45 Errata Document	Mask Set Errata for Mask P57D, Rev. 2.0

Chapter 3

Driver

- Requirements
- Driver Design Summary
- Hardware Resources
- Deviations from Requirements
- Driver Limitations
- Driver usage and configuration tips
- Runtime errors
- Symbolic Names Disclaimer

3.1 Requirements

PLATFORM is a complex driver, so there are no AUTOSAR requirements specific to this module. For the S32 platform, the PLATFORM module configures the interrupt controller, MCM and MSCM 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 and A53 cluster. The IsrHandler can be defined here or installed/updated at runtime via API.

Driver

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 S32, 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)

Linker files On S32, Internal RAM is different between these derivatives, linker file added more details information about memory size for each derivative. As currently, the linker has allocated memory for int_sram, int_sram ← _stack_, int_sram_no_cacheable, it is used for the common S32XX so that user can refer to allowcate for all derivatives.

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.

File Name	File Type	Description
startup.c Stub file. Must be replaced by all integrators.		This file is a stub. This file include sone func-
		tions necessary initializations for RAM. Copy
		the vector table from ROM to RAM. Copy ini-
		tialized data from ROM to RAM
exceptions.c	Stub file. Must be replaced by all integrators.	This file is a stub. This file necessary handler
		exception when running application.
startup_←	Stub file. Must be replaced by all integrators.	This file is a stub. This file need to run be-
cm7.s		fore initiatial application, it start-up code shall
		initialize the base addresses for interrupt and
		trap vector tables.
Vector_←	Stub file. Must be replaced by all integrators.	This file is a stub. This file necessary initial-
Table.s		izations for vector table.

3.3 Hardware Resources

#	Hardware IP	s32g274a	s32g254a	s32g233a	s32g234m	s32r45
1	NVIC	3	3	1	3	3
2	MSCM	1	1	1	1	1
3	MCM	3	3	1	3	3

3.4 Deviations from Requirements

None.

3.5 Driver Limitations

The PLATFORM driver software have some following limitations for RTD S32:

- Only one precompile configuration variant supported in the configuration tool
- No support for GPR registers

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, user should synchronize and disable cache before enable MPU, Platfom also supports sys_m7_cache_disable and sys_m7_cache_clean functions in system.c file so user can refer to it for your project, please note that MPU and Cache enablement in startup code is just demo code, user can find detail about MPU support in RM driver and Cache support in MCL driver. Here is some samples for configuration we can custom for startup:

Driver

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

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.

3.7 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 PlatformMcmConfigurable
 - * Parameter PlatformMscmConfigurable
 - * Parameter PlatformIpAPIsAvailable
 - * Parameter PlatformEnableUserModeSupport
 - * Parameter PlatformMulticoreSupport
 - * Parameter PlatformEnableVtorConfiguration
 - * Reference PlatformEcucPartitionRef
 - Container McmConfig
 - * Parameter SystemAhbSlavePrio
 - * Reference PlatformMcmEcucPartitionRef
 - * Container SystemIsrConfig
 - · Parameter SystemIsrName
 - · Parameter SystemIsrEnabled
 - Container IntCtrlConfig
 - * Parameter PlatformVtorAddressConfig
 - * Reference PlatformNvicEcucPartitionRef
 - * Container PlatformIsrConfig
 - · Parameter IsrName
 - · Parameter IsrEnabled
 - · Parameter IsrPriority
 - Container MscmConfig
 - * Reference PlatformGenericInterruptEcucPartitionRef
 - * Container PlatformIsrConfig
 - · Parameter IsrName
 - · Parameter IsrTargetCore0
 - · Parameter IsrTargetCore1

- · Parameter IsrTargetCore2
- · Parameter IsrTargetCore3
- · Parameter IsrTargetCore4
- · 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
- IntCtrlConfig
- MscmConfig
- 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 I2c 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

 ${\bf PlatformDevErrorDetect}$

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 PlatformMcmConfigurable

 ${\bf PlatformMcmConfigurable}$

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

4.5 Parameter PlatformMscmConfigurable

 ${\bf Platform Mscm Configurable}$

Check this in order to be able to configure Miscellaneous system 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 PlatformIpAPIsAvailable

 ${\bf Platform Ip AP Is Available}$

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

${\bf 4.7} \quad {\bf Parameter\ Platform Enable User Mode Support}$

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

4.8 Parameter PlatformMulticoreSupport

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

NoteThis is an	Implement	tation 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 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

4.10 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
l C £ Cl	VARIANT-POST-BUILD: PRE-COMPILE
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
${\it requires Symbolic Name Value}$	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.11 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

4.12 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']

4.13 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
${\it requires Symbolic Name Value}$	False
destination	/ AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.14 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
multiplicity ComigClasses	VARIANT-POST-BUILD: PRE-COMPILE

4.15 Parameter SystemIsrName

Vendor specific:

 ${\bf Interrupt\ Name.}$

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
varueComigClasses	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']

${\bf 4.16}\quad {\bf Parameter~System Isr Enabled}$

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
varueCollingClasses	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	false

4.17 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
multiplicity Colling Classes	VARIANT-POST-BUILD: PRE-COMPILE

${\bf 4.18} \quad {\bf Parameter\ PlatformVtorAddressConfig}$

Configure the address where the Interrupt Vector starts

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

4.19 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
	VARIANT-PRE-COMPILE: PRE-COMPILE
${\it requires Symbolic Name Value}$	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.20 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	195
upperMultiplicity	195
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.21 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
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MSCM_Pcie_1_MSI_IRQn

"MSCM_INT2_IRQn', 'MSCM_Pcie_0_MSI_IRQn', 'CTI_INT0_IRQn', TI_INT1_IRQn', 'MCM_IRQn', 'DMA0_0_15_IRQn', 'DMA0_16_3 RQn', 'DMA0_ERR0_IRQn', 'PMA1_0_15_IRQn', 'DMA1_6_31_IDMA1_ERR0_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', WT3_IRQn', 'SWT4_IRQn', 'SWT0_IRQn', 'SWT6_IRQn', 'SWT7_II'MSCM_INT3_IRQn', 'MSCM_INT4_IRQn', 'SWT6_IRQn', 'STM0_IRQn', 'STM1_II'STM2_IRQn', 'STM3_IRQn', 'STM4_IRQn', 'STM6_IRQn', 'STM1_II'STM2_IRQn', 'STM3_IRQn', 'STM4_IRQn', 'STM5_IRQn', 'STM6_IRQn', 'STM6_IRQn', 'STM6_IRQn', 'STM6_IRQn', 'STM6_IRQn', 'GAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ORED_IRQn', 'CAN1_ORED_IRQn', 'CAN1_ORED_IRQn', 'CAN1_ORED_IRQn', 'CAN2_ORED_B_127_MB_IRQn', 'CAN2_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'CAN3_ORED_B_127_MB_IRQn', 'GAN3_ORED_B_127_MB_IRQn', 'HSE_MU1_ARA_MB_IRQn', 'HSE_MU1_ARA_MB_IRQn', 'HSE_MU1_ARA_MB_IRQn', 'HSE_MU1_ARA_MB_IRQn', 'HSE_MU1_ARA_MB_IRQn', 'HSE_MU1_ARA_MB	Property	Value
"MSCM_INT2_IRQn," 'MSCM_Peie_0_MSI_IRQn," CTI_INTO_IRQn," TI_INT1_IRQn," 'MCM_IRQn," 'DMA0_0_15_ IRQn," 'DMA0_16_31_ IRQn," 'DMA0_ERRO_IRQn," 'DMA1_0_15_ IRQn," 'DMA0_16_31_ I'DMA1_ERRO_IRQn," 'SWTD_IRQn," 'SWD_IRQn," 'GND_IRQn," 'GND_IRQn," 'GND_IRQn," 'GND_IRQn," 'GND_IRQn," 'GND_ORED_IRQn," 'GND_OR	- •	['MSCM_Pcie_1_MSI_IRQn', 'MSCM_INT0_IRQn', 'MSCM_INT1_IRQn',
TI_INT1_IRQn', 'MCM_IRQn', 'DMA0_0_15_IRQn', 'DMA0_16_3 RQn', 'DMA0_ERG_IRQn', 'DMA0_16_15_IRQn', 'DMA0_16_15_IRQn', 'DMA0_16_15_IRQn', 'DMA0_16_15_IRQn', 'WT3_IRQn', 'SWT1_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', WT3_IRQn', 'SWT4_IRQn', 'SWT5_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT1_IRQ		'MSCM_INT2_IRQn', 'MSCM_Pcie_0_MSI_IRQn', 'CTI_INT0_IRQn', 'C
RQn', 'DMAD_ERRO_IRQn', 'SWT0_IRQn', 'GWT0_IRQn', 'GWT0_I		$\begin{array}{cccccccccccccccccccccccccccccccccccc$
'DMA1 ERRO IRQn', 'SWT0 IRQn', 'SWT1 IRQn', 'SWT7 IRQn', 'WT3 IRQn', 'WT5 IRQn', 'SWT6 IRQn', 'SWT7 I'MSCM_INT4 IRQn', 'STM6 IRQn', 'STM1 IRQn', 'STM2 IRQn', 'STM2 IRQn', 'STM4 IRQn', 'STM6 IRQn', 'STM5 IRQn', 'STM1 IRQn', 'STM6 IRQn', 'SP11 IRQn', 'SP12 IRQn', 'SAN0 ORED IRQn', 'CAN0 ORED IRQn', 'CAN0 ORED IRQn', 'CAN1 ORED IRQn', 'CAN1 ORED IRQn', 'CAN1 ORED IRQn', 'CAN1 ORED IRQn', 'CAN2 ORED IRQn', 'CAN3 ORED IRQn', 'GMA0 ORD IRQn', 'IRQn', 'IRQn', 'GMA0 ORD IRQn', 'IRQn', 'I		RQn', 'DMA0_ERR0_IRQn', 'DMA1_0_15_IRQn', 'DMA1_16_31_IRQn',
WT3_IRQp', 'SWT4_IRQp', 'SWT5_IRQp', 'SWT6_IRQp', 'STM_II' 'MSCM_INT3_IRQn', 'MSCM_INT4_IRQp', 'STM_IRQp', 'STM_III' 'STM2_IRQn', 'STM3_IRQn', 'STM4_IRQn', 'STM_IRQn', 'STM6_III' 'STM7_IRQn', 'QSP10_IRQn', 'QSP11_IRQn', 'QSP12_IRQn', 'STCU2 IST_MBIST_IRQn', 'USDHC_IRQn', 'QSP12_IRQn', 'STCU2 IST_MBIST_IRQn', 'USDHC_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ERD_IRQn', 'CAN0_ORED_0_7_ IRQn', 'CAN0_ORED_0_7_MB_IRQn', 'CAN0_ORED_8_127_MB_Qn', 'CAN1_ORED_IRQn', 'CAN1_ERR_IRQn', 'CAN0_ORED_0_7_ IRQn', 'CAN1_ORED_IRQn', 'CAN1_ERR_IRQn', 'CAN2_ORED_8_127_MB_IRQn', 'CAN2_ORED_8_127_MB_IRQn', 'CAN2_ORED_8_127_MB_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'CAN3_ORED_9_127_MB_IRQn', 'PST0_IRQn', 'PST0_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'PST0_IRQn', 'PST0_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'GMAC0_CH1_XX_IRQn', 'MSCM_INT5_IRQn',		'DMA1_ERR0_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', 'S
"MSCM_INT3_IRQn,', 'MSCM_INT4_IRQn,', 'STM5_IRQn,', 'STM6_II "STM2_IRQn,', 'STM3_IRQn,', 'STM4_IRQn,', 'STM5_IRQn,', 'STM6_II "STM7_IRQn,', 'QSP10_IRQn,', 'QSP10_IRQn,', 'QSP10_IRQn,', 'STCU_IST MBIST IRQn,', 'USDHC IRQn,', 'CANO_ORED_IRQn,', 'CANO_ORED_IRQn,', 'CANO_ORED_B_12T_MB_IRQn,', 'CANO_ORED_B_12T_MB_IRQn,', 'CAN_ORED_B_12T_MB_IRQn,', 'CMACO_CH_12T_IRQn,', 'CMACO_CH_B_12T_IRQn,', 'CMAC		WT3_IRQn', 'SWT4_IRQn', 'SWT5_IRQn', 'SWT6_IRQn', 'SWT7_IRQn',
"STM2 IRQn', 'STM3 IRQn', 'STM4 IRQn', 'STM5 IRQn', 'STM6 IR "STM7 IRQn', 'QSP10 IRQn', 'QSP11 IRQn', 'QSP12 IRQn', 'STM0 IST MBIST IRQn', 'USDMC IRQn', 'CAN0 ORED IRQn', 'CAN0 ORED IRQn', 'CAN1 ORED IRQn', 'CAN2 ORED REP MBQn', 'CAN2 ORED BER IRQn', 'CAN1 ORED SER IRQn', 'CAN2 ORED BER IRQn', 'CAN3 ORED IRQn', 'GMA0 ORED REP IRQn', 'GMA0 ORED SER IRQn', 'CAN3 ORED IRQn', 'GMA0 ORED IRQn', 'FIT1 ORQn', 'FIT1 ORQn', 'FIT1 ORQn', 'FIT1 ORQn', 'GMA0 ORED IRQn', 'FLEXRAY0 ORED IRQn', '		'MSCM_INT3_IRQn', 'MSCM_INT4_IRQn', 'STM0_IRQn', 'STM1_IRQn',
"STM_ IRQn', 'QSP10_ IRQn', 'QSP12_ IRQn', 'CAN0_ ORED_ IRQn', 'CAN0_ ERQn', 'CAN0_ ORED_ IRQn', 'CAN0_ ORED_ IRQn', 'CAN0_ ORED_ IRQn', 'CAN0_ ORED_ S 127_ MB Qn', 'CAN1_ ORED_ IRQn', 'CAN1_ ERR_ IRQn', 'CAN1_ ORED_ 0.7IRQn', 'CAN1_ ORED_ 8_127_ MB_ IRQn', 'CAN2_ ORED_ 1RQn', 'CAN2ERR_ IRQn', 'CAN2_ ORED_ 0.7_ MB_ IRQn', 'CAN2_ ORED_ 8_127_ B IRQn', 'CAN3_ ORED_ 1RQn', 'CAN3_ ERR_ IRQn', 'CAN3_ ORED_ 8_127_ MB_ IRQn', 'PIT0_ IRQn', 'PIT1_ Qn', 'FTM0_ IRQn', 'FTM1_ IRQn', 'GMAC0_ Common_ IRQn', 'PIT1_ Qn', 'FTM0_ IRQn', 'GMAC0_ CH0_ RX_ IRQn', 'GMAC0_ CH1_ TX_ IRQn', 'GMAC0_ CH1_ RX_ IRQn', 'GMAC0_ CH1_ TX_ IRQn', 'GMAC0_ CH1_ RX_ IRQn', 'FLEXRAY0_ CH1_ RX_ IRQn', 'GMAC0_ CH1_ RX_ IRQn', 'FLEXRAY0_ RX_ BUFF_		'STM2_IRQn', 'STM3_IRQn', 'STM4_IRQn', 'STM5_IRQn', 'STM6_IRQn',
IST MBIST IRQn', 'USDHC IRQn', 'CANO ORED IRQn', 'CANO ENQn', 'CANO ORED _ 127 MB		'STM7_IRQn', 'QSPI0_IRQn', 'QSPI1_IRQn', 'QSPI2_IRQn', 'STCU2_LB↔
RQn', 'CAN0_ ORED_0_7, 'MB_IRQn', 'CAN0_ ORED_8_127_ MB_Qn', 'CAN1_ ORED_IRQn', 'CAN1_ ORED_0_7_ RQn', 'CAN1_ ORED_1RQn', 'CAN2_ ORED_0_7_ RQn', 'CAN1_ ORED_1RQn', 'CAN2_ ORED_1RQn', 'CAN2_ ORED_1RQn', 'CAN2_ ORED_1RQn', 'CAN2_ ORED_1RQn', 'CAN2_ ORED_8_127_ MB_IRQn', 'CAN2_ ORED_8_127_ MB_IRQn', 'CAN3_ ORED_1RQn', 'CAN3_ ORED_8_127_ MB_IRQn', 'GMAC0_ CH0_RX_IRQn', 'GMAC0_ CH0_RX_IRQn', 'GMAC0_ CH0_RX_IRQn', 'GMAC0_ CH1_RX_IRQn', 'GMAC0_ CH2_RX_IRQn', 'GMAC0_ CH2_RX_IRQn', 'GMAC0_ CH2_RX_IRQn', 'GMAC0_ CH3_RX_IRQn', 'GMAC0_ CH3_RX_IRQn', 'GMAC0_ CH3_RX_IRQn', 'GMAC0_ CH3_RX_IRQn', 'GMAC0_ CH4_RX_IRQn', 'MAC0_ CH4_RX_IRQn', 'MAC0_ CH3_RX_IRQn', 'GMAC0_ CH4_RX_IRQn', 'MAC0_ CH3_RX_IRQn', 'GMAC0_ CH3_RX_IRQn', 'FIEXRAY0_ CMBERR_II 'FIEXRAY0_ TX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'FIEXRAY0_RX_IRQn', 'SP10_IRQn', 'SP11_IRQn', 'SP12_IRQn', 'SP13_IRQn', 'FIEXRAY0_RX_BUFF_IRQn', 'F		
Qn', 'CAN1_ORED_ IRQn', 'CAN1_ERR_ IRQn', 'CAN2_ORED_ IRQn', 'CAN2_ERR_ IRQn', 'CAN2_ORED_ 8_127_ MB_ IRQn', 'CAN2_ORED_ 8_127_ B_ IRQn', 'CAN2_ORED_ 8_127_ B_ IRQn', 'CAN3_ORED_ 7_ MB_ IRQn', 'CAN3_ORED_ 7_ MB_ IRQn', 'CAN3_ORED_ 7_ MB_ IRQn', 'CAN3_ORED_ 8_127_ MB_ IRQn', 'PITO_ 1RQn', 'PITO_ 1RQn', 'PITO_ 1RQn', 'PITO_ 1RQn', 'ITMO_ IRQn', 'ITMO_ IRQn', 'ITMO_ IRQn', 'ITMO_ IRQn', 'GMACO_CH0_RX_ IRQn', 'GMACO_CH1_TX_ IRQn', 'GMACO_CH1_TX_ IRQn', 'GMACO_CH2_TX_ IRQn', 'GMACO_CH2_TX_ IRQn', 'GMACO_CH3_TX_ IRQn', 'GMACO_CH3_TX_ IRQn', 'GMACO_CH3_TX_ IRQn', 'GMACO_CH3_TX_ IRQn', 'GMACO_CH3_TX_ IRQn', 'GMACO_CH4_TX_ IRQn', 'FIEXRAYO_CH1_RX_ FIFO_IRQn', 'FIEXRAYO_CH1_RX_ FIFO_IRX_ FIFO_IRX_ FIFO_IRX_ FIFO_IRX_ FIFO_IRX_ FIFO_IRX_ FIFO_IRX_		_IRQn', 'CAN0_ORED_0_7_MB_IRQn', 'CAN0_ORED_8_127_MB_IR
ERR IRQn', 'CAN2_ORED_ IRQn', 'CAN2_ORED_ 8_127 B_IRQn', 'CAN3_ORED_ IRQn', 'CAN3_ERR_IRQn', 'CAN3_ORED_ T_MB_IRQn', 'CAN3_ORED_ 8_127_MB_IRQn', 'PIT1 IRQn', 'PIT1 IRQn', 'FTM0_IRQn', 'FTM1_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH1_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'MSCM_INT5_IRQn', 'ELEXRAY0_CH1_RX_FIF0_IRQn', 'FLEXRAY0_CH1_RX_FIF0_IRQn', 'FLEXRAY0_CH1_RX_FIF0_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'TMSCM_INT5_IRQn', 'MSE_IRQn', 'MSE_IRQn', 'MSE_IRQn', 'MSE_IRQn', 'MSE_IRQn', 'MSE_MU1_TX_IRQn', 'MSE_MU1_TX_I		Qn', 'CAN1_ORED_IRQn', 'CAN1_ERR_IRQn', 'CAN1_ORED_0_7_MB↔
B_IRQn', 'CAN3_ORED_ RQn', 'CAN3_ERR_IRQn', 'CAN3_ORED_ 7 MB_IRQn', 'CAN3_ORED_ 8 127 MB_IRQn', 'PITI_Qn', 'FTM0_IRQn', 'FITM1_IRQn', 'GMAC0_Cmmon_IRQn', 'GMAC0_H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'MAC0_CH1_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'MAC0_CH1_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'GMAC0_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'INT5_IRQn', 'MSCM_INT5_IRQn', 'INT5_IRQN', '		_IRQn', 'CAN1_ORED_8_127_MB_IRQn', 'CAN2_ORED_IRQn', 'CAN2
7_MB_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'PIT0_IRQn', 'PIT1_Qn', 'FTM0_IRQn', 'FTM1_IRQn', 'GMACO_Common_IRQn', 'GMACO_H0_TX_IRQn', 'GMACO_CH0_RX_IRQn', 'GMACO_CH1_TX_IRQn', 'MACO_CH2_TX_IRQn', 'GMACO_CH2_R IRQn', 'GMACO_CH3_TX_IRQn', 'GMACO_CH3_RX_IRQn', 'GMACO_CH3_RX_IRQn', 'GMACO_CH4_RX_IRQn', 'GMACO_CH3_RX_IRQn', 'GMACO_H4_TX_IRQn', 'GMACO_CH3_RX_IRQn', 'GMACO_H4_TX_IRQn', 'GMACO_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MS_INT6_IRQn', 'GMACO_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MS_INT6_IRQn', 'GMACO_CH4_RX_IRQn', 'SAR_ADC1_INT_IRQn', 'INT6_IRQn', 'FIEXRAYO_CH4_RX_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CERR_IRQn', 'FIEXRAYO_CMBERR_III_FIEXRAYO_CMBERR_III_FIEXRAYO_TX_BUFF_IRQn', 'FIEXRAYO_RX_BUFF_IRQn', 'FIEXRAYO_CMBERR_III_FIEXRAYO_TX_BUFF_IRQn', 'FIEXRAYO_RX_BUFF_IRQn', 'FIEXRAYO_RX_BUFF_IRQn', 'FIEXRAYO_TX_BUFF_IRQn', 'FIEXRAYO_RX_BUFF_IRQn', 'FIEXRAYO_TX_BUFF_IRQn', 'HSE_MU_TX_IRQn', 'FIEXRAYA_CUNTATANO_TX_IRQn', 'PCIEO_INTA_IRQn', 'PCIEO_INTA_IRQn', 'PCIEO_INTA_IRQn', 'PCIEO_INTA_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'SWTIB_IRQn', 'ILCEO_INTA_IRQn', 'LLCEO_INTA_IRQn', 'LLCEO_INTA_IRQn', 'LLCEO_INTA_I		_ERR_IRQn', 'CAN2_ORED_0_7_MB_IRQn', 'CAN2_ORED_8_127_M
Qn', 'FTM0_IRQn', 'FTM1_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', MAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH4_TX_IRQn', 'GMAC0_CH4_TX_IRQn', 'GMAC0_CH4_TX_IRQn', 'GMAC0_CH3_TX_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'TEXRAY0_CH2_TX_IRQn', 'FLEXRAY0_CH2_TX_IRQn', 'FLEXRAY0_CH2_TX_IRQn', 'FLEXRAY0_CH3_TX_IRQn', 'FLEXRAY0_CMBERN_INT5_IRQn', 'FLEXRAY0_TX_IRQn', 'FLEXRAY0_TX_IRQn', 'FLEXRAY0_TX_IRQn', 'TEXPACA_TX_IRQn', 'TEXPACA_T		B_IRQn', 'CAN3_ORED_IRQn', 'CAN3_ERR_IRQn', 'CAN3_ORED_0_
H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'MAC0_CH2_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH2_RX_IRQn', 'GMAC0_CH3_RX_IRQn', 'GMAC0_CH3_RX_IRQn', 'GMAC0_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'FIXRAY0_CERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CH1_RX_FIFO_IRQn', 'FLEXRAY0_CMBERR_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'FLEXD2_IRQn', 'SP10_IRQn', 'SP11_IRQn', 'SP12_IRQn', 'SP12_IRQn', 'SP12_IRQn', 'SP12_IRQn', 'ISP12_IRQn', 'ISP14_IRQn', 'ILCE0_INT14_IRQn', '		7_MB_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'PIT0_IRQn', 'PIT1_IR
MACO_CH1_RX_IRQn', 'GMACO_CH2_TX_IRQn', 'GMACO_CH2_R IRQn', 'GMACO_CH3_TX_IRQn', 'GMACO_CH3_TX_IRQn', 'GMACO_CH4_TX_IRQn', 'GMACO_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'FINT6_IRQn', 'SAR_ADCO_INT_IRQn', 'FINT6_IRQn', 'SAR_ADCO_INT_IRQn', 'SAR_ADCO_INT_IRQn', 'FINT6_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'FINT6_IRQn', 'FIN		Qn', 'FTM0_IRQn', 'FTM1_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_C↔
MACO_CH1_RX_IRQn', 'GMACO_CH2_TX_IRQn', 'GMACO_CH2_R IRQn', 'GMACO_CH3_TX_IRQn', 'GMACO_CH3_TX_IRQn', 'GMACO_CH4_TX_IRQn', 'GMACO_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'MSCM_INT5_IRQn', 'FINT6_IRQn', 'SAR_ADCO_INT_IRQn', 'FINT6_IRQn', 'SAR_ADCO_INT_IRQn', 'SAR_ADCO_INT_IRQn', 'FINT6_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'SPID_IRQn', 'FINT6_IRQn', 'FIN		H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'G↔
H4_TX_IRQn', 'GMAC0_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MS_INT6_IRQn', 'SAR_ADC0_INT_IRQn', 'SAR_ADC1_INT_IRQn', 'TXRAY0_NCERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CERR_X_FIFO_IRQn', 'FLEXRAY0_CHI_RX_FIFO_IRQn', 'FLEXRAY0_CM_RX_FIFO_IRQn', 'FLEXRAY0_CMBERR_II 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_CMBERR_II 'FLEXRAY0_TX_BUFF_IRQn', 'INFLEXD0_IRQn', 'INFLEXD1_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_TX_BUFF_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn', 'FLEXD2_IRQn', 'SP10_IRQn', 'SP10_IRQn', 'SP10_IRQn', 'SP12_IRQn', 'SP12_IRQn', 'SP13_IRQn', 'FLAXO_IRQn', 'SP15_IRQn', 'ISC0_IRQn', 'ISC1_IRQn', 'ISC2_IRQn', 'II_IRQn', 'ISE3_IRQn', 'ISE3_I		MAC0_CH1_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_
INT6_IRQn', 'SAR_ADC0_INT_IRQn', 'SAR_ADC1_INT_IRQn', 'FXRAY0_NCERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CHR_X_FIFO_IRQn', 'FLEXRAY0_CHR_X_FIFO_IRQn', 'FLEXRAY0_CHR_X_FIFO_IRQn', 'FLEXRAY0_CHR_X_FIFO_IRQn', 'FLEXRAY0_CMBERR_IR_FIEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'SPIG_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'FLEXD2_IRQn', 'SPIG_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'FLIRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCCM_INTD_IRQn', 'FCCCM_INTD_IRQn', 'FCCCM_INTD_IRQn', 'FCCCM_INTD_IRQn', 'FCCCM_INTD_IRQn', 'FCCCC_ALARM, 'FCC		IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH3_RX_IRQn', 'GMAC0_C↔
XRAY0_NCERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CERX_FIFO_IRQn', 'FLEXRAY0_CHI_RX_FIFO_IRQn', 'FLEXRAY0_KUP_IRQn', 'FLEXRAY0_STATUS_IRQn', 'FLEXRAY0_CMBERR_I' 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn', 'FLEXD2_IRQn', 'SPI5_IRQn', 'IZC1_IRQn', 'IZC2_IRQn', 'FLEXD2_IRQn', 'SPI5_IRQn', 'IZC0_IRQn', 'FLCU_ALARM_IRQn', 'FCC_MISC_IRQn', 'IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCC_MISC_IRQn', 'ISSBW_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RX_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU1_X_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU1_X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_CERD_IRQn', 'TMU_ALARM_IRQn', 'FCTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIBO_ORED_DMA_IRQn', 'PCIBO_INTC_IRQn', 'PCIBO_INTA_IRQn', 'PCIBO_DMA_IRQn', 'PCIBO_INTC_IRQn', 'PCIBO_INTA_IRQn', 'PCIBO_INTB_IRQn', 'PCIBO_INTA_IRQn', 'PCIBO_INTB_IRQn', 'PCIBO_INTC_IRQn', 'PCIBO_INTA_IRQn', 'PCIBO_INTS_IRQn', 'PCIBO_INTA_IRQn', 'PCIBO_INTS_IRQn', 'PCIBO_INTS_IRQn', 'PCIBO_INTS_IRQn', 'PCIBO_INTS_IRQn', 'PCIBO_INTS_IRQn', 'SWTS_IRQn', 'STMS_IRQn', 'MCSCM_INTS_IRQn', 'ILCEO_INTS_IRQn', 'LLCEO_INTS_IRQn', 'L		H4_TX_IRQn', 'GMAC0_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM↔
RX_FIFO_IRQn', 'FLEXRAY0_CH1_RX_FIFO_IRQn', 'FLEXRAY0_KUP_IRQn', 'FLEXRAY0_CMBERR_II 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FL RAY0_MODULE_IRQn', 'LINFLEXDO_IRQn', 'LINFLEXD1_IRQn', 'FL RAY0_MODULE_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn' PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I: _IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I: _IRQn', 'ISES_MU0_ORED_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_M RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'FRO_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_INY_IRQn', 'PCIE0_A MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_INTC_IRQn', 'PC INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_INTC_IRQn', 'PC INTD_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'SWT5_IRQn', 'STM10_IRQn', 'SWT10_IRQn', 'STM10_IRQn', 'SWT10_IRQn', 'SWT5_IRQn', 'STM10_IRQn', 'SWT10_IRQn', 'SWT5_IRQn', 'STM10_IRQn', 'STM11_IRQn', 'CSCM_INT7_IRQn', 'MCSCM_INT5_IRQn', 'ILCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'LCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'ILCEO_INT5_IRQn', 'LLCEO_INT5_IRQn', 'LLCEO_INT5_IRQ		_INT6_IRQn', 'SAR_ADC0_INT_IRQn', 'SAR_ADC1_INT_IRQn', 'FLE↔
KUP_IRQn', 'FLEXRAY0_STATUS_IRQn', 'FLEXRAY0_CMBERR_II 'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'IL RAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'IF FLEXD2_IRQn', 'SPI0_IRQn', 'ISPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn' PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2 _IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FC _MISC_IRQn', 'SBSW_IRQn', 'HSE_MU_TX_IRQn', 'HSE_MU_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU2_ORED_IRQn', 'BDRO_SCRUB_IRQn', 'FC _MX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDRO_SCRUB_IRQn', 'RO_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'PCIE0_AMSI_IRQn', 'PCIE0_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_ _MSI_IRQn', 'PCIE0_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_LINTC_IRQn', 'PCIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'SWTS_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT _RCLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWTS_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'IMCSCM_INT9_IRQn', 'SCCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'LICE0_INT0_IRQn', 'LICE0 _CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0 _CSR14_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0 _CSR14_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0 _ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0 _ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0 _ICSR18_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0		XRAY0_NCERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CH0_
'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FL RAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'J FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn' PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2 IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2 IRQn', 'SBSW_IRQn', 'ISE_MU0_TX_IRQn', 'ISE_MU0_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_M RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'PCIE0_IRQn', 'PCIE0_IRQn', 'PCIE0_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_IRQn', 'PCIE0		RX_FIFO_IRQn', 'FLEXRAY0_CH1_RX_FIFO_IRQn', 'FLEXRAY0_W
RAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn', PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I1_IRQn', 'I2C4_IRQn', 'I2C0_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2C		KUP_IRQn', 'FLEXRAY0_STATUS_IRQn', 'FLEXRAY0_CMBERR_IRQn',
FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn' PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCM_ALARM_IRQn', 'FCCM_ALARM_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RED_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_AMSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', 'CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_TINC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTD_IRQn', 'CORTE_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT3_IRQn', 'SWT3_IRQn', 'SWT10_IRQn', 'SWT3_IRQn', 'STM10_IRQn', 'SWT3_IRQn', 'STM10_IRQn', 'SWT3_IRQn', 'STM10_IRQn', 'TSM10_IRQn', 'SCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_ICSR16_IRQn', 'ILCE0_INT3_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0_ICSR18_IRQn', 'ILCE0_ICSR19_IRQn', 'ILCE0		'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEX↔
PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I5C0_IRQn', 'I5C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCU_MISC_IRQn', 'SBSW_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT IRQn', 'STM8_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' IRQn', 'STM8_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'ILCE0_INT3_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_I		RAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'LIN←
IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCMISC_IRQn', 'SBSW_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MRX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSEMU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_COIRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_AMSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn'CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL_NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'CORT_A53_ERR_LVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_LVLOCK_CLUSTER0_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'STM11_IRQn', 'STM11_IRQn', 'STM10_IRQn', 'STM11_IRQn', 'CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR19_IRQn', 'LC		FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn', 'S
_MISC_IRQn', 'SBSW_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_R IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_M _RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_M _RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'B0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO _IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A _MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PC _INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'SWT10_IRQn', 'CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'INCSCM_INT9_IRQn', 'ICCSCM_INT9_IRQn', 'ICCSCM_INT11_IRQn', 'ULCE0_INT3_IRQn', 'ICCSCN14_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INS17_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR19_IRQn', 'ILCE0_ICSR19_IRQn', 'ILC		$\begin{tabular}{ l l l l l l l l l l l l l l l l l l l$
IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1 X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_M RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_M MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', ' R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO _IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A _MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_NTC_IRQn', 'PC _INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'SWT10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT8_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'L		_IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCU↔
X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_M RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_REL D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PC INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_IN		_MISC_IRQn', 'SBSW_IRQn', 'HSE_MU0_TX_IRQn', 'HSE_MU0_RX_
RX_IRQn', 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A_MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', 'PCIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_PHY_UP_IRQn', 'PCIE0_INTA_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL_NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'CORTA_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORTA_A53_ERR_L1VLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L1VLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT8_IRQn', 'STM8_IRQn', 'STM10_IRQn', 'STM11_IRQn', 'CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'ILCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_		IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_R↔
MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A_MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', 'CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL_NC_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'CORT_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'GORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT9_IRQn', 'STM11_IRQn', 'GSCM_INT7_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', 'CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'ICSCM_INT10_IRQn', 'ICSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT11_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR18_IRQn', 'LLC		
R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELD_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A_MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', 'CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCIE0_INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL_NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW'_IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'IMCSCM_INT9_IRQn', 'ICOR_INT0_IRQn', 'ICOR_INT11_IRQn', 'LLCE0_INT0_IRQn', 'ICOR_INT11_IRQn', 'LLCE0_INT11_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQ		$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CO _IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_A _MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PC _INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICS		MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'DD↔
IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_AMSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PCINTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORTA53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR19_		R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELOA -
_MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn' CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PC _INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'L E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0 CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'L		
CIE0_INTA_IRQn', 'PCIE0_INTB_IRQn', 'PCIE0_INTC_IRQn', 'PC _INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TL NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1 Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'L E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_I		
INTDIRQn', 'PCIE0MISCIRQn', 'PCIE0PCSIRQn', 'PCIE0TL NCIRQn', 'CORTEXA53ERRL2RAMCLUSTER0IRQn', 'CORTA53ERRLIVLOCKCLUSTER0IRQn', 'CORTEXA53ERRL2 MCLUSTER1IRQn', 'CORTEXA53ERRLIVLOCKCLUSTER1_ Qn', 'JDCIRQn', 'SWT8IRQn', 'SWT9IRQn', 'SWT10IRQn', 'SW'IRQn', 'STM8IRQn', 'STM9IRQn', 'STM10IRQn', 'STM11IRQn', CSCMINT7IRQn', 'MCSCMINT8IRQn', 'MCSCMINT9IRQn', 'I SCMINT10IRQn', 'MCSCMINT11IRQn', 'LLCE0INT0IRQn', 'L E0INT1IRQn', 'LLCE0INT2IRQn', 'LLCE0INT3IRQn', 'LLCE0CSR14IRQn', 'LLCE0ICSR15IRQn', 'LLCE0ICSR16IRQn', 'LLCE0ICSR17IRQn', 'LLCE0ICSR18IRQn', 'LLCE0ICSR19IRQn', 'LCE0ICSR19IRQn', 'LLCE0ICSR19IRQn', 'LLCE0IC		
NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORT _A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_ Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE1_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE1_ICSR17_IRQn', 'LLCE1_ICSR18_IRQn', 'LLCE1_ICSR19_IRQn', 'LLCE1_ICSR19		
_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2 M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_ Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'I SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'I		
M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'IDCSCM_INT11_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'IDCSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'IDCSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'IDCSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'IDCSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'IDCSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_I		
Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SW' _IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE1_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE1_ICSR19_IRQn', 'LCE1_ICSR19_IRQn', 'LLCE1_ICSR19_IRQn', 'LLCE1_ICSR1		
IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LLCE0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICS		
CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LEO_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LCE0_ICSR19_IRQn', 'LLCE0_ICSR19_IRQn', 'LLCE0_ICSR1		
SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'I E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLC _ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'I		
E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0 CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLC _ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'I		
CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLC _ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'I		
_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'I		
TEM MORZO TRUM, TATARA MORZI TRUM, TATARA IOSB22 TRUM,		
		CE0_ICSR23_IRQn', 'LLCE0_ICSR24_IRQn', 'LLCE0_ICSR25_IRQn', 'L
		LCEO_ICSR26_IRQn', 'LLCEO_ICSR27_IRQn', 'PFE0_CH0_STAT_IRQn', 'PFE0_CH1_STAT_IRQn', 'PFE
NXP Semiconductors PFEU_CHI_STAT_IRQn', 'PFEU_CH2_STAT_IRQn', 'PFEU_CH3_S NXP Semiconductors T_IDOn! S32; PLATFORM Driver IDEED_HIE_NC_IDON! IDEED	IXP Semiconductors	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$
T_indi, Treo_biot_biot_indi, Treo_inr_io_indi, Tr		$\begin{array}{cccccccccccccccccccccccccccccccccccc$
		OI_GPI_IRQn', 'PFE0_PMI_IRQn', 'PFE0_ORED_IRQn', 'SIM_IS_ CH_REO_IROn', 'SIMLI_ORED_IROn', 'USBO_OTG_CORE_IROn', 'II.,

Property	Value
----------	-------

4.22 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

4.23 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	15
min	0

4.24 Container MscmConfig

Generic configuration for the interrupts (routing, handlers).

Included subcontainers:

• PlatformIsrConfig

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

${\bf 4.25} \quad {\bf Reference\ Platform Generic Interrupt Ecuc Partition Ref}$

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
varueComigCiasses	VARIANT-PRE-COMPILE: PRE-COMPILE
${\it requires Symbolic Name Value}$	False
destination	/AUTOSAR/EcucDefs/EcuC/EcucPartitionCollection/EcucPartition

4.26 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	195
upperMultiplicity	195
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A

4.27 Parameter IsrName

Vendor specific:

Interrupt Name.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
${\it symbolicNameValue}$	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
varueComigCiasses	VARIANT-POST-BUILD: PRE-COMPILE
defaultValue	MSCM_Pcie_1_MSI_IRQn

Property	Value
literals	['MSCM_Pcie_1_MSI_IRQn', 'MSCM_INT0_IRQn', 'MSCM_INT1_IRQn',
	'MSCM_INT2_IRQn', 'MSCM_Pcie_0_MSI_IRQn', 'CTI_INT0_IRQn', 'C
	TI_INT1_IRQn' , 'MCM $_IRQn'$, 'DMA0 $_0_15_IRQn'$, 'DMA0 $_16_31_I$
	RQn', 'DMA0_ERR0_IRQn', 'DMA1_0_15_IRQn', 'DMA1_16_31_IRQn',
	'DMA1_ERR0_IRQn', 'SWT0_IRQn', 'SWT1_IRQn', 'SWT2_IRQn', 'S
	WT3_IRQn', 'SWT4_IRQn', 'SWT5_IRQn', 'SWT6_IRQn', 'SWT7_IRQn',
	'MSCM_INT3_IRQn', 'MSCM_INT4_IRQn', 'STM0_IRQn', 'STM1_IRQn',
	'STM2_IRQn', 'STM3_IRQn', 'STM4_IRQn', 'STM5_IRQn', 'STM6_IRQn',
	'STM7_IRQn', 'QSPI0_IRQn', 'QSPI1_IRQn', 'QSPI2_IRQn', 'STCU2_LB↔
	IST_MBIST_IRQn', 'USDHC_IRQn', 'CAN0_ORED_IRQn', 'CAN0_ERR←
	_IRQn', 'CAN0_ORED_0_7_MB_IRQn', 'CAN0_ORED_8_127_MB_IR←
	Qn', 'CAN1_ORED_IRQn', 'CAN1_ERR_IRQn', 'CAN1_ORED_0_7_MB↔
	_IRQn', 'CAN1_ORED_8_127_MB_IRQn', 'CAN2_ORED_IRQn', 'CAN2←
	_ERR_IRQn', 'CAN2_ORED_0_7_MB_IRQn', 'CAN2_ORED_8_127_M↔
	B_IRQn', 'CAN3_ORED_IRQn', 'CAN3_ERR_IRQn', 'CAN3_ORED_0_
	7_MB_IRQn', 'CAN3_ORED_8_127_MB_IRQn', 'PIT0_IRQn', 'PIT1_IR↔
	Qn', 'FTM0_IRQn', 'FTM1_IRQn', 'GMAC0_Common_IRQn', 'GMAC0_C↔
	H0_TX_IRQn', 'GMAC0_CH0_RX_IRQn', 'GMAC0_CH1_TX_IRQn', 'G↔
	MAC0_CH1_RX_IRQn', 'GMAC0_CH2_TX_IRQn', 'GMAC0_CH2_RX_
	IRQn', 'GMAC0_CH3_TX_IRQn', 'GMAC0_CH3_RX_IRQn', 'GMAC0_C↔
	H4_TX_IRQn', 'GMAC0_CH4_RX_IRQn', 'MSCM_INT5_IRQn', 'MSCM↔
	_INT6_IRQn', 'SAR_ADC0_INT_IRQn', 'SAR_ADC1_INT_IRQn', 'FLE
	XRAY0_NCERR_IRQn', 'FLEXRAY0_CERR_IRQn', 'FLEXRAY0_CH0_
	RX_FIFO_IRQn', 'FLEXRAY0_CH1_RX_FIFO_IRQn', 'FLEXRAY0_W
	KUP_IRQn', 'FLEXRAY0_STATUS_IRQn', 'FLEXRAY0_CMBERR_IRQn',
	'FLEXRAY0_TX_BUFF_IRQn', 'FLEXRAY0_RX_BUFF_IRQn', 'FLEX↔
	RAY0_MODULE_IRQn', 'LINFLEXD0_IRQn', 'LINFLEXD1_IRQn', 'LIN←
	FLEXD2_IRQn', 'SPI0_IRQn', 'SPI1_IRQn', 'SPI2_IRQn', 'SPI3_IRQn', 'S
	PI4_IRQn', 'SPI5_IRQn', 'I2C0_IRQn', 'I2C1_IRQn', 'I2C2_IRQn', 'I2C3←
	_IRQn', 'I2C4_IRQn', 'MC_RGM_IRQn', 'FCCU_ALARM_IRQn', 'FCCU←
	$_MISC_IRQn', \ 'SBSW_IRQn', \ 'HSE_MU0_TX_IRQn', \ 'HSE_MU0_RX_{\leftarrow}$
	IRQn', 'HSE_MU0_ORED_IRQn', 'HSE_MU1_TX_IRQn', 'HSE_MU1_R↔
	X_IRQn', 'HSE_MU1_ORED_IRQn', 'HSE_MU2_TX_IRQn', 'HSE_MU2~
	$RX_{IRQn'}$, 'HSE_MU2_ORED_IRQn', 'HSE_MU3_TX_IRQn', 'HSE_ \leftarrow
	MU3_RX_IRQn', 'HSE_MU3_ORED_IRQn', 'DDR0_SCRUB_IRQn', 'DD↔
	R0_PHY_IRQn', 'CTU_FIFO_FULL_EMPTY_IRQn', 'CTU_M_RELOA↔
	D_IRQn', 'CTU_ERR_IRQn', 'TMU_ALARM_IRQn', 'RTC_SYS_CONT
	_IRQn', 'PCIE0_ORED_DMA_IRQn', 'PCIE0_LINK_IRQn', 'PCIE0_AXI↔
	_MSI_IRQn', 'PCIE0_PHY_DOWM_IRQn', 'PCIE0_PHY_UP_IRQn', 'P↔
	CIEO_INTA_IRQn', 'PCIEO_INTB_IRQn', 'PCIEO_INTC_IRQn', 'PCIEO↔
	_INTD_IRQn', 'PCIE0_MISC_IRQn', 'PCIE0_PCS_IRQn', 'PCIE0_TLP_
	NC_IRQn', 'CORTEX_A53_ERR_L2RAM_CLUSTER0_IRQn', 'CORTEX-
	_A53_ERR_LIVLOCK_CLUSTER0_IRQn', 'CORTEX_A53_ERR_L2RA-
	M_CLUSTER1_IRQn', 'CORTEX_A53_ERR_LIVLOCK_CLUSTER1_IR
	Qn', 'JDC_IRQn', 'SWT8_IRQn', 'SWT9_IRQn', 'SWT10_IRQn', 'SWT11\Laplace
	_IRQn', 'STM8_IRQn', 'STM9_IRQn', 'STM10_IRQn', 'STM11_IRQn', 'M \
	CSCM_INT7_IRQn', 'MCSCM_INT8_IRQn', 'MCSCM_INT9_IRQn', 'MC
	SCM_INT10_IRQn', 'MCSCM_INT11_IRQn', 'LLCE0_INT0_IRQn', 'LLC
	E0_INT1_IRQn', 'LLCE0_INT2_IRQn', 'LLCE0_INT3_IRQn', 'LLCE0_I CONTAINTS IRQn', 'LLCE0_I CONTAINTS IRQn', 'LLCE0_ICONTAINTS IRQn', 'LLCE0_ICONTAINT
	CSR14_IRQn', 'LLCE0_ICSR15_IRQn', 'LLCE0_ICSR16_IRQn', 'LLCE0-
	_ICSR17_IRQn', 'LLCE0_ICSR18_IRQn', 'LLCE0_ICSR19_IRQn', 'LLC
	E0_ICSR20_IRQn', 'LLCE0_ICSR21_IRQn', 'LLCE0_ICSR22_IRQn', 'LLC
	CE0_ICSR23_IRQn', 'LLCE0_ICSR24_IRQn', 'LLCE0_ICSR25_IRQn', 'L
	LCEO_ICSR26_IRQn', 'LLCEO_ICSR27_IRQn', 'PFE0_CH0_STAT_IRQn',
NXP Semiconductors	'PFE0_CH1_STAT_IRQn', 'PFE0_CH2_STAT_IRQn', 'PFE0_CH3_STA↔ T_IRQn', 'S32; PLATFQRM, Priveron', IREE0_HE_NC_IRQn', 'PFE0_CH3_STA↔ 27
	T_IRQn', S32 PLATFORM Driveron', 'PFE0_HIF_NC_IRQn', 'PFE0_27
	UT_GPT_IRQn', 'PFE0_PMT_IRQn', 'PFE0_ORED_IRQn', 'STM_TS_ CH_REO_IROn', 'SHILL_ORED_IROn', 'USBO_OTG_CORE_IROn', 'U.
	THE RED TRUM' SHILL DRED TRUM' TISED OTC CORE TROM' THA

Property	Value
----------	-------

${\bf 4.28}\quad {\bf Parameter~IsrTargetCore0}$

Vendor specific:

Select the target core for the interrupt request. Parameter is readonly if this target core is not available.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
${\it symbolicNameValue}$	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

4.29 Parameter IsrTargetCore1

Vendor specific:

Select the target core for the interrupt request. Parameter is readonly if this target core is not available.

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

${\bf 4.30}\quad {\bf Parameter\ Isr Target Core 2}$

Vendor specific:

Select the target core for the interrupt request. Parameter is readonly if this target core is not available.

Property	Value
type	ECUC-BOOLEAN-PARAM-DEF
origin	NXP
${\it symbolicNameValue}$	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

${\bf 4.31} \quad {\bf Parameter~IsrTargetCore3}$

Vendor specific:

Select the target core for the interrupt request. Parameter is readonly if this target core is not available.

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

4.32 Parameter IsrTargetCore4

Vendor specific:

Select the target core for the interrupt request. Parameter is readonly if this target core is not available.

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

4.33 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

4.34 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

4.35 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
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	4
max	4
min	4

4.36 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

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

4.37 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
${\it symbolicNameValue}$	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

4.38 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

Property	Value
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	255
max	255
min	255

4.39 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
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	4
max	4
min	4

4.40 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

Tresos Configuration Plug-in

Property	Value
max	0
min	0

4.41 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
${\it symbolicNameValue}$	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
varueComigCiasses	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	0
min	0

4.42 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

Property	Value
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	

4.43 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
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:

Platform	4	18
Interrupt Controller IP		37
System IP	5	57

Chapter 6

Module Documentation

6.1 Interrupt Controller IP

6.1.1 Detailed Description

Data Structures

- $\bullet \ \ struct \ IntCtrl_Ip_IrqRouteConfigType \\$
 - Structure storing the routing and handler configuration for an interrupt request. More...
- $\bullet \ \ struct \ IntCtrl_Ip_GlobalRouteConfigType$
 - Structure storing the list of routing configurations for all configured interrupts. More...
- 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.

• void IntCtrl_Ip_ClearDirectedCpuInterrupt (IRQn_Type eIrqNumber)

Clear directed cpu Interrupt interrupt flag.

- boolean IntCtrl_Ip_GetDirectedCpuInterrupt (IRQn_Type eIrqNumber)
 - Get directed cpu Interrupt interrupt flag.
- void IntCtrl_Ip_GenerateDirectedCpuInterrupt (IRQn_Type eIrqNumber, IntCtrl_Ip_IrqTargetType e ← CpuTarget)

Generates an interrupt request to a CPU target.

6.1.2 Data Structure Documentation

6.1.2.1 struct IntCtrl_Ip_IrqRouteConfigType

Structure storing the routing and handler configuration for an interrupt request.

Definition at line 75 of file IntCtrl Ip TypesDef.h.

Data Fields

• IRQn_Type eIrqNumber

 $Interrupt\ number.$

• uint8 u8TargetCores

Target cores for the interrupt.

• IntCtrl Ip IrqHandlerType pfHandler

Interrupt handler.

6.1.2.1.1 Field Documentation

$6.1.2.1.1.1 \quad eIrqNumber \quad \textit{IRQn_Type eIrqNumber}$

Interrupt number.

Definition at line 78 of file IntCtrl Ip TypesDef.h.

6.1.2.1.1.2 u8TargetCores uint8 u8TargetCores

Target cores for the interrupt.

Definition at line 80 of file IntCtrl_Ip_TypesDef.h.

6.1.2.1.1.3 pfHandler IntCtrl_Ip_IrqHandlerType pfHandler

Interrupt handler.

Definition at line 82 of file IntCtrl_Ip_TypesDef.h.

6.1.2.2 struct IntCtrl_Ip_GlobalRouteConfigType

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

Definition at line 89 of file IntCtrl_Ip_TypesDef.h.

Data Fields

- uint32 u32ConfigIrqCount
 - Number of configured interrupts.
- const IntCtrl_Ip_IrqRouteConfigType * aIrqConfig

 $List\ of\ interrupts\ configurations.$

6.1.2.2.1 Field Documentation

$\mathbf{6.1.2.2.1.1} \quad u32 Config Irq Count \quad \texttt{uint32} \ u32 \texttt{Config Irq} \texttt{Count}$

Number of configured interrupts.

Definition at line 92 of file IntCtrl_Ip_TypesDef.h.

$\mathbf{6.1.2.2.1.2} \quad \mathbf{aIrqConfig} \quad \mathtt{const} \; \; \mathtt{IntCtrl_Ip_IrqRouteConfigType*} \; \; \mathtt{aIrqConfig}$

List of interrupts configurations.

Definition at line 94 of file IntCtrl_Ip_TypesDef.h.

6.1.2.3 struct IntCtrl_Ip_IrqConfigType

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

Definition at line 101 of file IntCtrl_Ip_TypesDef.h.

Data Fields

- IRQn_Type eIrqNumber

 Interrupt number.
- boolean bIrqEnabled

Interrupt state (enabled/disabled)

• uint8 u8IrqPriority

Interrupt priority.

6.1.2.3.1 Field Documentation

6.1.2.3.1.1 eIrqNumber IRQn_Type eIrqNumber

Interrupt number.

Definition at line 104 of file IntCtrl_Ip_TypesDef.h.

6.1.2.3.1.2 bIrqEnabled boolean bIrqEnabled

Interrupt state (enabled/disabled)

Definition at line 106 of file IntCtrl_Ip_TypesDef.h.

6.1.2.3.1.3 u8IrqPriority uint8 u8IrqPriority

Interrupt priority.

Definition at line 108 of file IntCtrl Ip TypesDef.h.

6.1.2.4 struct IntCtrl_Ip_CtrlConfigType

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

Definition at line 115 of file IntCtrl_Ip_TypesDef.h.

Data Fields

- uint32 u32ConfigIrqCount
 - Number of configured interrupts.
- const IntCtrl_Ip_IrqConfigType * aIrqConfig

List of interrupts configurations.

6.1.2.4.1 Field Documentation

6.1.2.4.1.1 u32ConfigIrqCount uint32 u32ConfigIrqCount

Number of configured interrupts.

Definition at line 118 of file IntCtrl_Ip_TypesDef.h.

6.1.2.4.1.2 aIrqConfig const IntCtrl_Ip_IrqConfigType* aIrqConfig

List of interrupts configurations.

Definition at line 124 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 69 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 131 of file IntCtrl_Ip_TypesDef.h.

$\bf 6.1.4.2 \quad IntCtrl_Ip_IrqTargetType$

```
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.
INTCTRL_IP_TARGET_CP3	Interrupt request targeted to core 3.
INTCTRL_IP_TARGET_CP4	Interrupt request targeted to core 4.
INTCTRL_IP_TARGET_CP5	Interrupt request targeted to core 5.
INTCTRL_IP_TARGET_CP6	Interrupt request targeted to core 6.

Definition at line 170 of file IntCtrl_Ip_TypesDef.h.

6.1.5 Function Reference

6.1.5.1 IntCtrl_Ip_Init()

Initializes the configured interrupts at interrupt controller level.

This function is non-reentrant and initializes the interrupts.

Parameters

in	pIntCtrlCtrlCcnfig	pointer to configuration structure for interrupts.
----	--------------------	--

Returns

 $IntCtrl_Ip_StatusType: error code.$

6.1.5.2 IntCtrl_Ip_InstallHandler()

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()

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()

Disables an interrupt request.

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

Parameters

Returns

void.

6.1.5.5 IntCtrl_Ip_SetPriority()

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()

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 ( {\tt IRQn\_Type}\ eIrq{\tt Number}\ )
```

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.

6.1.5.8 IntCtrl_Ip_ClearDirectedCpuInterrupt()

Clear directed cpu Interrupt interrupt flag.

This function is non-reentrant; it is provided for clearing directed cpu Interrupt interrupt flag.

Parameters

Returns

void.

6.1.5.9 IntCtrl_Ip_GetDirectedCpuInterrupt()

```
boolean IntCtrl_Ip_GetDirectedCpuInterrupt (  \label{eq:int} {\tt IRQn\_Type} \ eIrqNumber \ )
```

Get directed cpu Interrupt interrupt flag.

This function is non-reentrant; it is provided for getting directed cpu Interrupt interrupt flag.

Parameters

in	eIrqNumber	interrupt number
----	------------	------------------

Returns

boolean: TRUE - flag set, FALSE - flag cleared.

6.1.5.10 IntCtrl_Ip_GenerateDirectedCpuInterrupt()

Generates an interrupt request to a CPU target.

This function is non-reentrant; it is provided for generating a directed interrupt to a CPU defined by target parameter.

Parameters

in	eIrqNumber	interrupt number to be triggered.
in	eCpuTarget	target core for the interrupt request.

Returns

void.

6.2 Platform

6.2.1 Detailed Description

Modules

- Interrupt Controller 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_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.

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.

6.2.2 Data Structure Documentation

6.2.2.1 struct Platform ConfigType

Configuration structure for PLATFORM CDD.

Definition at line 175 of file Platform TypesDef.h.

Data Fields

- const Platform_Ipw_ConfigType * pIpwConfig Reference to IPW structure.
- const Platform_Ipw_NonCoreConfigType * pIpwNonCoreConfig Reference to Core Independent IPW structure.

6.2.2.1.1 Field Documentation

$\mathbf{6.2.2.1.1.1} \quad \mathbf{pIpwConfig} \quad \mathtt{const} \; \; \mathtt{Platform_Ipw_ConfigType*} \; \; \mathtt{pIpwConfig}$

Reference to IPW structure.

Definition at line 178 of file Platform_TypesDef.h.

6.2.2.1.1.2 pIpwNonCoreConfig const Platform_Ipw_NonCoreConfigType* pIpwNonCoreConfig

Reference to Core Independent IPW structure.

Definition at line 180 of file Platform_TypesDef.h.

6.2.3 Macro Definition Documentation

6.2.3.1 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.

6.2.3.2 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.

6.2.3.3 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.

6.2.3.4 PLATFORM_INIT_ID

#define PLATFORM_INIT_ID

Service ID of Platform Init function.

Parameter used when raising an error/exception

Definition at line 106 of file Platform_TypesDef.h.

$\bf 6.2.3.5 \quad PLATFORM_SET_IRQ_ID$

#define PLATFORM_SET_IRQ_ID

Service ID of Platform_SetIrq function.

Parameter used when raising an error/exception

Definition at line 112 of file Platform TypesDef.h.

6.2.3.6 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 118 of file Platform_TypesDef.h.

6.2.3.7 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 124 of file Platform_TypesDef.h.

6.2.3.8 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 130 of file Platform_TypesDef.h.

6.2.3.9 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 136 of file Platform_TypesDef.h.

6.2.3.10 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 142 of file Platform_TypesDef.h.

6.2.3.11 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 148 of file Platform_TypesDef.h.

6.2.3.12 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 154 of file Platform_TypesDef.h.

6.2.3.13 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 160 of file Platform_TypesDef.h.

6.2.3.14 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 166 of file Platform_TypesDef.h.

6.2.4 Types Reference

6.2.4.1 Platform_IrqHandlerType

```
typedef IntCtrl_Ip_IrqHandlerType Platform_IrqHandlerType
```

Interrupt handler type definition for PLATFORM CDD.

Definition at line 187 of file Platform_TypesDef.h.

6.2.5 Function Reference

6.2.5.1 Platform_Init()

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

6.2.5.2 Platform_SetIrq()

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.

6.2.5.3 Platform_SetIrqPriority()

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.

6.2.5.4 Platform_GetIrqPriority()

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.

6.2.5.5 Platform_InstallIrqHandler()

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.

6.3 System IP

6.3.1 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.

6.3.2 Function Reference

6.3.2.1 System_Ip_SetAhbSlavePriority()

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

6.3.2.2 System_Ip_ConfigIrq()

 ${\bf Enables/disables~core\text{-}related~interrupt~exceptions.}$

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

6.3.2.3 System_Ip_ClearWriteAbortFlag()

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

6.3.2.4 System_Ip_GetPlatformRevision()

```
uint32 System_Ip_GetPlatformRevision ( void \quad )
```

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

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.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorlQ, QorlQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2022 NXP B.V.

