

User Manual

for S32K1 RM Driver

Document Number: UM2RMASR4.4 Rev0000R1.0.1 Rev. 1.0

1 Revision History	2
2 Introduction	3
2.1 Supported Derivatives	3
2.2 Overview	4
2.3 About This Manual	5
2.4 Acronyms and Definitions	6
2.5 Reference List	6
3 Driver	7
3.1 Requirements	7
3.2 Driver Design Summary	7
3.2.1 MPU	8
3.3 Hardware Resources	8
3.4 Deviations from Requirements	9
3.5 Driver Limitations	9
3.6 Driver usage and configuration tips	9
3.6.1 HLD usage	9
3.6.2 LLD usage	9
3.6.3 Protection of QSPI memory region requires 2 Region Descriptors (Only for S32K148)	10
3.7 Runtime errors	10
3.8 Symbolic Names Disclaimer	10
4 Tresos Configuration Plug-in	11
4.1 Module Rm	12
4.2 Container RmGeneral	12
4.3 Parameter Rm_VersionInfoApi	13
4.4 Parameter RmDevErrorDetect	13
4.5 Parameter RmEnableUserModeSupport	14
4.6 Parameter RmMpuConfigurable	14
4.7 Container RmConfigSet	15
4.8 Container Mpu_Configuration	15
4.9 Parameter MpuDevErrorDetect	15
4.10 Container MpuRegionConfig	16
4.11 Parameter RegionNumber	16
4.12 Parameter StartAddress	17
4.13 Parameter EndAddress	17
4.14 Parameter ProcessIdentifierEnableMaster0	18
4.15 Parameter SupervisorModeAccessBusMaster0	18
4.16 Parameter UserModeAccessBusMaster0	19
4.17 Parameter ProcessIdentifierEnableMaster1	19

4.18 Parameter SupervisorModeAccessBusMaster1	20
4.19 Parameter UserModeAccessBusMaster1	20
4.20 Parameter SupervisorModeAccessBusMaster2	21
4.21 Parameter UserModeAccessBusMaster2	21
4.22 Parameter SupervisorModeAccessBusMaster3	22
4.23 Parameter UserModeAccessBusMaster3	22
4.24 Parameter ProcessIdentifier	23
4.25 Parameter ProcessIdentifierMask	23
4.26 Container CommonPublishedInformation	24
4.27 Parameter ArReleaseMajorVersion	24
4.28 Parameter ArReleaseMinorVersion	25
4.29 Parameter ArReleaseRevisionVersion	25
4.30 Parameter ModuleId	26
4.31 Parameter SwMajorVersion	26
4.32 Parameter SwMinorVersion	27
4.33 Parameter SwPatchVersion	27
4.34 Parameter VendorApiInfix	29
4.35 Parameter VendorId	30
5 Module Index	31
5.1 Software Specification	31
6 Module Documentation	32
6.1 RM Driver	32
6.1.1 Detailed Description	32
6.1.2 Data Structure Documentation	33
6.1.3 Enum Reference	34
6.1.4 Function Reference	36
6.1.5 Variable Documentation	38
6.2 MPU IPV Driver	39
6.2.1 Detailed Description	39
6.2.2 Function Reference	39



Chapter 1

Revision History

Revision	Date	Author	Description
1.0	24.02.2022	NXP RTD Team	Prepared for release RTD S32K1 Version 1.0.1

Chapter 2

Introduction

- [Supported Derivatives](#)
- [Overview](#)
- [About This Manual](#)
- [Acronyms and Definitions](#)
- [Reference List](#)

This User Manual describes Resource Manager for S32K1. Resource Manager driver configuration parameters and deviations from the specification are described in Driver chapter of this document.

2.1 Supported Derivatives

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

- s32k116_qfn32
- s32k116_lqfp48
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k142_lqfp64
- s32k142_lqfp100
- s32k142w_lqfp48
- s32k142w_lqfp64
- s32k144_lqfp48
- s32k144_lqfp64

- s32k144_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100
- s32k146_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176

All of the above microcontroller devices are collectively named as S32K1.

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
C/CPP	C and C++ Source Code
CDD	Complex Device Driver
DEM	Diagnostic Event Manager
DET	Development Error Tracer
ECU	Electronic Control Unit
LSB	Least Significant Bit
MCU	Micro Controller Unit
MIDE	Multi Integrated Development Environment
MPU	Memory Protection Unit
MRC	Memory Region Controller
MSB	Most Significant Bit
N/A	Not Applicable
RAM	Random Access Memory
RM	Resource Manager
SIU	Systems Integration Unit
SWS	Software Specification
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	S32K1 Series Reference Manual	Rev. 14, 09/2021
2	S32K116_0N96V	Rev. 22/OCT/2021
3	S32K118_0N97V	Rev. 22/OCT/2021
4	S32K142_0N33V	Rev. 22/OCT/2021
5	S32K144_0N57U	Rev. 22/OCT/2021
6	S32K144W_0P64A	Rev. 22/OCT/2021
7	S32K146_0N73V	Rev. 22/OCT/2021
8	S32K148_0N20V	Rev. 22/OCT/2021
9	S32K1xx Data Sheet	Rev. 14, 08/2021



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

Resource Manager is a Complex Device Driver (CDD), so there are no AUTOSAR requirements regarding this module.

It has vendor-specific requirements and implementation.

3.2 Driver Design Summary

The RM module provides a way to initialize and control the resource domains allocation and memory protection on the chip with the supported peripherals.

3.2.1 MPU

MPU has 8 region descriptors (16 region descriptors for S32K148) configurable to protect memory (Flash, Ram, Peripheral memory) from access of masters (Core, Debugger, DMA, ENET). Each master can be authorized with 2 different access modes: user mode and supervisor mode.

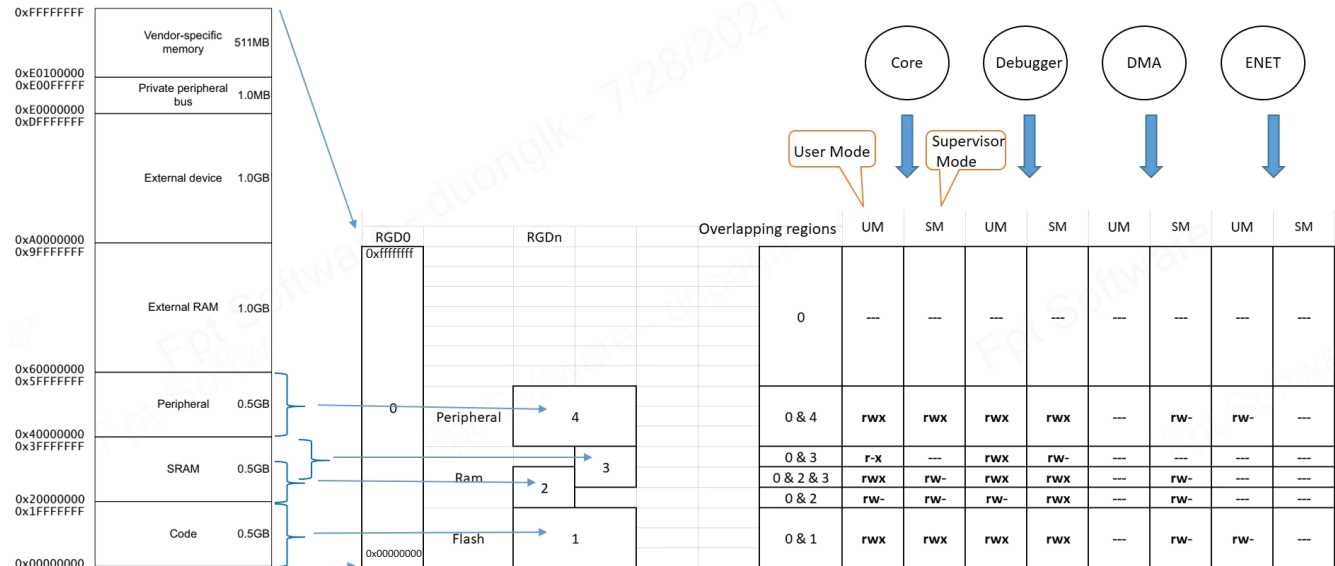


Figure 3.1 Overlapping region descriptor example

MPU Region Configuration									
Ind...	Name	Region Nu...	Start Address	End Address	Proce...	Supervisor Mode Access Bus Master0 (Core)	User Mode Access Bus Master0 (Core)	Proce...	Super
0	MpuRegionConfig_0	0	0x0	0xffffffff		MPU_SUPERVISOR_MODE_AS_USER_MODE	MPU_USER_MODE_NONE		MPU_Si
1	Flash	1	0x0	0x1bffffff		MPU_SUPERVISOR_MODE_RWX	MPU_USER_MODE_RWX		MPU_Si
2	SRAM_L0	2	0x1c000000	0x1ffffff		MPU_SUPERVISOR_MODE_RW	MPU_USER_MODE_RW		MPU_Si
3	SRAM_L1_U	3	0x20000000	0x200ffff		MPU_SUPERVISOR_MODE_RX	MPU_USER_MODE_NONE		MPU_Si
4	Peripheral	4	0x40000000	0x5ffffff		MPU_SUPERVISOR_MODE_RWX	MPU_USER_MODE_RWX		MPU_Si

Figure 3.2 Example of MPU configuration interface on the EB tresos

The lowest 5 bits of the start address of a region descriptor are always 0. The start address must be configured to be divisible by 32. The lowest 5 bits of the end address of a region descriptor are always 1. The end address must be configured to divide 32 with remainder 0x1f.

Overlapping region descriptors: Region descriptors may overlap. The protection rights of overlapping memory areas is logically summed from region descriptors together (boolean OR operator).

Region descriptor 0 (Region Number=0) must cover all memory (0x00000000-0xffffffff).

3.3 Hardware Resources

#	Hardware IP	Description
1	MPU	Memory Protection Unit

3.4 Deviations from Requirements

Since this is a CDD Module, there are no AUTOSAR requirements for the functionality.

3.5 Driver Limitations

- None

3.6 Driver usage and configuration tips

3.6.1 HLD usage Prior usage of the RM CDD in an application, the configuration files must be generated with the configurator.

Initialization of all memory regions of MPU is done using [Rm_Init](#). Considering that the function is configuring masters and access to memory regions it is recommended that it is called before other bus masters are active, or that they are not accessing those memory regions.

The rest of the API controls directly the allocated hardware resources. Check the references below for more information:

- [Rm_Mpu_SetRegionConfig](#)
- [Rm_Mpu_EnableRegion](#)
- [Rm_Mpu_SetAccessMode](#)
- [Rm_Mpu_GetErrorDetails](#)

3.6.2 LLD usage Prior usage of the LLDs from RM CDD in an application, the configuration files must be generated with the S32CT configurator.

For more information on the LLD API check the references below:

- [MPU IPV Driver](#)
 - [Mpu_Ip_Init](#)
 - [Mpu_Ip_SetRegionConfig](#)
 - [Mpu_Ip_Deinit](#)
 - [Mpu_Ip_EnableRegion](#)
 - [Mpu_Ip_SetAccessMode](#)
 - [Mpu_Ip_GetErrorDetails](#)

3.6.3 Protection of QSPI memory region requires 2 Region Descriptors (Only for S32K148)

The MPU requires a special programming sequence to protect the QSPI space as it is unable to see the two MSB bits of the QSPI address on slave port 4. This programming sequence requires 2 Region Descriptors [RGDx].

Workaround: In order to properly provide protection to the QSPI space, it is necessary to use 2 RGDs. One will cover the region 0x280x_xxxx and the other one will cover region 0x680x_xxxx. When any master without permissions tries to access region 0x680x_xxxx, an error will be captured in both, EDR3 and EDR4 registers. Moreover, the address of the failed access is captured on EAR3 and EAR4 registers. However, EAR3 will capture the address 0x680x_xxxx, which is the one that belongs to the QSPI space. While EAR4 will capture the 0x280x_xxxx address.

MPU Region Configuration*

Ind...	Name	Region Nu...	Start Address	End Address	Proce...	Supervisor Mode Access Bus Master...	User Mode Access Bus Maste.
0	MpuRegionConfig_0	0	0x0	0xffffffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_NONE
1	MpuRegionConfig_1	1	0x0	0x1bffffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_RWX
2	MpuRegionConfig_2	2	0x1c000000	0x200fffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_RWX
3	MpuRegionConfig_3	3	0x40000000	0x400fffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_RWX
4	QSPI_SlavePort_4	4	0x28000000	0x280fffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_RWX
5	QSPI_SlavePort_3	5	0x68000000	0x680fffff		MPU_SUPERVISOR_MODE_AS_USER...	MPU_USER_MODE_RWX

Figure 3.3 Protection of QSPI memory region requires 2 Region Descriptors

3.7 Runtime errors

- Development Error Description

Error Code	Value	Condition triggering the error
RM_E_INIT_FAILED_U8	4	Rm_Init() is called with input parameter is invalid
RM_E_ALREADY_INITIALIZED_U8	2	Rm_Init() is called when the RM module is being initialized
RM_E_PARAM_POINTER	5	Rm_GetVersionInfo() is called with input parameter is NULL_PTR

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 [Rm](#)
 - Container [RmGeneral](#)
 - * Parameter [Rm_VersionInfoApi](#)
 - * Parameter [RmDevErrorDetect](#)
 - * Parameter [RmEnableUserModeSupport](#)
 - * Parameter [RmMpuConfigurable](#)
 - Container [RmConfigSet](#)
 - * Container [Mpu_Configuration](#)
 - Parameter [MpuDevErrorDetect](#)
 - Container [MpuRegionConfig](#)
 - Parameter [RegionNumber](#)
 - Parameter [StartAddress](#)
 - Parameter [EndAddress](#)
 - Parameter [ProcessIdentifierEnableMaster0](#)
 - Parameter [SupervisorModeAccessBusMaster0](#)
 - Parameter [UserModeAccessBusMaster0](#)
 - Parameter [ProcessIdentifierEnableMaster1](#)
 - Parameter [SupervisorModeAccessBusMaster1](#)
 - Parameter [UserModeAccessBusMaster1](#)
 - Parameter [SupervisorModeAccessBusMaster2](#)
 - Parameter [UserModeAccessBusMaster2](#)
 - Parameter [SupervisorModeAccessBusMaster3](#)
 - Parameter [UserModeAccessBusMaster3](#)
 - Parameter [ProcessIdentifier](#)
 - Parameter [ProcessIdentifierMask](#)
 - Container [CommonPublishedInformation](#)
 - * Parameter [ArReleaseMajorVersion](#)
 - * Parameter [ArReleaseMinorVersion](#)
 - * Parameter [ArReleaseRevisionVersion](#)

- * Parameter [ModuleId](#)
- * Parameter [SwMajorVersion](#)
- * Parameter [SwMinorVersion](#)
- * Parameter [SwPatchVersion](#)
- * Parameter [VendorApiInfix](#)
- * Parameter [VendorId](#)

4.1 Module Rm

Vendor specific: Configuration of the Rm (Resource Manager) module.

Included containers:

- [RmGeneral](#)
- [RmConfigSet](#)
- [CommonPublishedInformation](#)

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

4.2 Container RmGeneral

Vendor specific: Configuration of general Rm parameters.

Included subcontainers:

- None

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

4.3 Parameter Rm_VersionInfoApi

Vendor specific: Enables/Disables the get version info API function

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-POST-BUILD: PRE-COMPILE
	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	true

4.4 Parameter RmDevErrorDetect

Vendor specific:

Switches the Development Error Detection and Notification on or off.

true: Enabled.

false: Disabled.

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

4.5 Parameter RmEnableUserModeSupport

When this parameter is enabled, the RM 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.

for more information, please see chapter 5.7 User Mode Support in IM

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 RmMpuConfigurable

RmMpuM7Configurable

Check this in order to be able to use the MPU.

Note: Implementation Specific Parameter.

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

4.7 Container RmConfigSet

Vendor specific: This container is the base for a multiple configuration set

Included subcontainers:

- [Mpu_Configuration](#)

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

4.8 Container Mpu_Configuration

Configuration for the MPU module.

Included subcontainers:

- [MpuRegionConfig](#)

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

4.9 Parameter MpuDevErrorDetect

MpuDevErrorDetect

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

4.10 Container MpuRegionConfig

Vendor specific:

Configuration for Mpu regions

Included subcontainers:

- None

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

4.11 Parameter RegionNumber

Vendor specific:

Hardware Region Number to be configuration.

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

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

4.12 Parameter StartAddress

Vendor specific:

Start Address of the region.

Note: For RegionNumber 0, Start address always is 0x00000000, This field will be ignored.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	0
max	4294967264
min	0

4.13 Parameter EndAddress

Vendor specific:

End Address of the region.

Note: For RegionNumber 0, End address always is 0xFFFFFFFF, This field will be ignored.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
default Value	31
max	4294967295
min	31

4.14 Parameter ProcessIdentifierEnableMaster0

Bus Master 0 (Core) Process Identifier enable.

True: Include the process identifier and mask in the region hit evaluation.

False: Do not include the process identifier in the evaluation.

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

4.15 Parameter SupervisorModeAccessBusMaster0

Vendor specific: Defines the access controls for bus master 0 in Supervisor mode.

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

4.16 Parameter UserModeAccessBusMaster0

Vendor specific: Defines the access controls for bus master 0 in User mode.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MPU_USER_MODE_NONE
literals	['MPU_USER_MODE_NONE', 'MPU_USER_MODE_RWX', 'MPU_USER_MODE_RW', 'MPU_USER_MODE_RX', 'MPU_USER_MODE_R', 'MPU_USER_MODE_WX', 'MPU_USER_MODE_W', 'MPU_USER_MODE_X']

4.17 Parameter ProcessIdentifierEnableMaster1

Bus Master 1 (Debugger) Process Identifier enable.

Tresos Configuration Plug-in

True: Include the process identifier and mask in the region hit evaluation.

False: Do not include the process identifier in the evaluation.

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

4.18 Parameter SupervisorModeAccessBusMaster1

Vendor specific: Defines the access controls for bus master 1 in Supervisor mode.

Note: Can not set supervisor mode access bus master 1 for region 0

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

4.19 Parameter UserModeAccessBusMaster1

Vendor specific: Defines the access controls for bus master 1 in User mode.

Note: Can not set user mode access bus master 1 for region 0

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MPU_USER_MODE_NONE
literals	['MPU_USER_MODE_NONE', 'MPU_USER_MODE_RWX', 'MPU_USER_MODE_RW', 'MPU_USER_MODE_RX', 'MPU_USER_MODE_R', 'MPU_USER_MODE_WX', 'MPU_USER_MODE_W', 'MPU_USER_MODE_X']

4.20 Parameter SupervisorModeAccessBusMaster2

Vendor specific: Defines the access controls for bus master 2 in Supervisor mode.

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

4.21 Parameter UserModeAccessBusMaster2

Vendor specific: Defines the access controls for bus master 2 in User mode.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MPU_USER_MODE_NONE
literals	['MPU_USER_MODE_NONE', 'MPU_USER_MODE_RWX', 'MPU_USER_MODE_RW', 'MPU_USER_MODE_RX', 'MPU_USER_MODE_R', 'MPU_USER_MODE_WX', 'MPU_USER_MODE_W', 'MPU_USER_MODE_X']

4.22 Parameter SupervisorModeAccessBusMaster3

Vendor specific: Defines the access controls for bus master 3 in Supervisor mode.

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

4.23 Parameter UserModeAccessBusMaster3

Vendor specific: Defines the access controls for bus master 3 in User mode.

Property	Value
type	ECUC-ENUMERATION-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	true
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
	VARIANT-POST-BUILD: POST-BUILD
defaultValue	MPU_USER_MODE_NONE
literals	['MPU_USER_MODE_NONE', 'MPU_USER_MODE_RWX', 'MPU_USER_MODE_R_MODE_RW', 'MPU_USER_MODE_RX', 'MPU_USER_MODE_R', 'MPU_USER_MODE_WX', 'MPU_USER_MODE_W', 'MPU_USER_MODE_X']

4.24 Parameter ProcessIdentifier

Vendor specific: Specifies the process identifier that is included in the region hit determination

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

4.25 Parameter ProcessIdentifierMask

Vendor specific: Provides a masking capability so that multiple process identifiers can be included as part of the region hit determination

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

4.26 Container CommonPublishedInformation

Vendor specific:

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

4.27 Parameter ArReleaseMajorVersion

Vendor specific:

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: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.28 Parameter ArReleaseMinorVersion

Vendor specific:

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	4
max	4
min	4

4.29 Parameter ArReleaseRevisionVersion

Vendor specific:

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

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

4.30 Parameter ModuleId

Vendor specific:

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: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	255
max	255
min	255

4.31 Parameter SwMajorVersion

Vendor specific:

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: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	1
max	1
min	1

4.32 Parameter SwMinorVersion

Vendor specific:

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: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.33 Parameter SwPatchVersion

Vendor specific:

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

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

4.34 Parameter VendorApiInfix

Vendor specific:

requires In driver modules which can be instantiated several times on a single ECU, BSW00347

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>_<VendorAp

E.g. assuming that the VendorId of the implementor is 123 and the implementer chose a VendorApiInfix of "v11r456" a api name Can_Write defined in the SWS will translate to Can_123_v11r456Write.

This parameter is mandatory for all modules with upper multiplicity > 1. It shall not be used for modules with upper multiplicity =1.

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

S32K1 RM Driver

4.35 Parameter VendorId

Vendor specific:

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: PUBLISHED-INFORMATION
	VARIANT-POST-BUILD: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43



Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

RM Driver	32
MPU IPV Driver	39

Chapter 6

Module Documentation

6.1 RM Driver

6.1.1 Detailed Description

Data Structures

- struct [Mpu_Ip_RegionConfigType](#)
Configuration structure containing the region configuration. [More...](#)
- struct [Mpu_Ip_ConfigType](#)
IP configuration structure. [More...](#)
- struct [Mpu_Ip_ErrorDetailsType](#)
Structure used to retrieve violation details. [More...](#)

Enum Reference

- enum [Mpu_Ip_SupervisorAccessModeType](#)
Enumeration listing access permissions in supervisor mode.
- enum [Mpu_Ip_UserAccessModeType](#)
Enumeration listing access permissions in user mode.
- enum [Mpu_Ip_MasterType](#)
Enumeration listing masters.
- enum [Mpu_Ip_ErrorAttributesType](#)
Structure used to retrieve error attributes details.
- enum [Mpu_Ip_AccessType](#)
Structure used to retrieve error attributes details.

Function Reference

- void [Rm_Init](#) (Rm_ConfigType const *ConfigPtr)
This function initializes the RM hardware components.
- void [Rm_Mpu_SetRegionConfig](#) (uint8 u8RegionNum, const [Rm_Mpu_RegionConfigType](#) *const pUserConfigPtr)
Configures the region selected by u8RegionNum with the data from pUserConfigPtr.
- void [Rm_Mpu_EnableRegion](#) (uint8 u8RegionNum, boolean bEnable)
Enables or disabled a specific region.
- void [Rm_Mpu_SetAccessMode](#) (uint8 u8RegionNum, [Rm_Mpu_MasterType](#) eMaster, [Rm_Mpu_SupervisorAccessModeType](#) eSupervisorMode, [Rm_Mpu_UserAccessModeType](#) eUserMode)
Modify the access mode for a master to a specific region.
- Std_ReturnType [Rm_Mpu_GetErrorDetails](#) ([Rm_Mpu_ErrorDetailsType](#) *pErrorDetails)
Retrieve error details.

Variables

- const Rm_ConfigType [Rm_Config](#)

6.1.2 Data Structure Documentation

6.1.2.1 struct Mpu_Ip_RegionConfigType

Configuration structure containing the region configuration.

Definition at line 100 of file Mpu_Ip_TypesDef.h.

Data Fields

Type	Name	Description
uint32	u32StartAddr	Memory region start address - Word 0
uint32	u32EndAddr	Memory region end address - Word 1
uint32	u32Word2	Access permission for region - Word 2
uint32	u32Pid	Process Identifier - Word 3
uint32	u32PidMask	Process Identifier Mask - Word 3

6.1.2.2 struct Mpu_Ip_ConfigType

IP configuration structure.

Definition at line 112 of file Mpu_Ip_TypesDef.h.

Data Fields

Type	Name	Description
uint8	u8RegionCnt	Region Count
const Mpu_Ip_RegionConfigType *	pRegionConfigArr	Region configuration array
const uint8 *	pRegionNumberArr	Region Number array

6.1.2.3 struct Mpu_Ip_ErrorDetailsType

Structure used to retrieve violation details.

Definition at line 142 of file Mpu_Ip_TypesDef.h.

Data Fields

Type	Name	Description
uint32	u32Address	Violation address
uint16	u16EACD	Error Access Control Detail
Mpu_Ip_MasterType	eMaster	Violation master
Mpu_Ip_ErrorAttributesType	eErrorAttribute	Type of Attribute violation
Mpu_Ip_AccessType	eErrorAccess	Type of Access violation

6.1.3 Enum Reference

6.1.3.1 Mpu_Ip_SupervisorAccessModeType

enum [Mpu_Ip_SupervisorAccessModeType](#)

Enumeration listing access permissions in supervisor mode.

Enumerator

MPU_SUPERVISOR_MODE_RWX	0b00U : rwx
MPU_SUPERVISOR_MODE_RX	0b01U : r-x
MPU_SUPERVISOR_MODE_RW	0b10U : rw-
MPU_SUPERVISOR_MODE_AS_USER_MODE	0b11U : —

Definition at line 62 of file Mpu_Ip_TypesDef.h.

6.1.3.2 Mpu_Ip_UserAccessModeType

enum `Mpu_Ip_UserAccessModeType`

Enumeration listing access permissions in user mode.

Enumerator

MPU_USER_MODE_NONE	0b000U : —
MPU_USER_MODE_X	0b001U : -x
MPU_USER_MODE_W	0b010U : -w
MPU_USER_MODE_WX	0b011U : -wx
MPU_USER_MODE_R	0b100U : r-
MPU_USER_MODE_RX	0b101U : r-x
MPU_USER_MODE_RW	0b110U : rw-
MPU_USER_MODE_RWX	0b111U : rwx

Definition at line 74 of file `Mpu_Ip_TypesDef.h`.

6.1.3.3 Mpu_Ip_MasterType

enum `Mpu_Ip_MasterType`

Enumeration listing masters.

Definition at line 88 of file `Mpu_Ip_TypesDef.h`.

6.1.3.4 Mpu_Ip_ErrorAttributesType

enum `Mpu_Ip_ErrorAttributesType`

Structure used to retrieve error attributes details.

Definition at line 122 of file `Mpu_Ip_TypesDef.h`.

6.1.3.5 Mpu_Ip_AccessType

enum `Mpu_Ip_AccessType`

Structure used to retrieve error attributes details.

Definition at line 133 of file `Mpu_Ip_TypesDef.h`.

6.1.4 Function Reference

6.1.4.1 Rm_Init()

```
void Rm_Init (
    Rm_ConfigType const * ConfigPtr )
```

This function initializes the RM hardware components.

This service is a non reentrant function used for driver initialization. The Initialization function shall initialize all relevant registers of the configured hardware with the values of the structure referenced by the parameter ConfigPtr. If the hardware allows for only one usage of the register, the driver module implementing that functionality is responsible for initializing the register. The initialization function of this module shall always have a pointer as a parameter, even though for Variant PC no configuration set shall be given. Instead a NULL pointer shall be passed to the initialization function.

Parameters

in	<i>ConfigPtr</i>	Pointer to a selected configuration structure.
----	------------------	------------------------------------------------

Returns

void

6.1.4.2 Rm_Mpu_SetRegionConfig()

```
void Rm_Mpu_SetRegionConfig (
    uint8 u8RegionNum,
    const Rm_Mpu_RegionConfigType *const pUserConfigPtr )
```

Configures the region selected by u8RegionNum with the data from pUserConfigPtr.

This function is Reentrant

Parameters

in	<i>u8RegionNum</i>	region number
in	<i>pUserConfigPtr</i>	pointer to the region configuration structure for MPU.

Returns

void

Precondition

6.1.4.3 Rm_Mpu_EnableRegion()

```
void Rm_Mpu_EnableRegion (
    uint8 u8RegionNum,
    boolean bEnable )
```

Enables or disabled a specific region.

This function is Reentrant

Parameters

in	<i>u8Region</i>	: Region to be modified
in	<i>bEnable</i>	: Specifies whether the region is enabled or disabled

Returns

void

Precondition

6.1.4.4 Rm_Mpu_SetAccessMode()

```
void Rm_Mpu_SetAccessMode (
    uint8 u8RegionNum,
    Rm_Mpu_MasterType eMaster,
    Rm_Mpu_SupervisorAccessModeType eSupervisorMode,
    Rm_Mpu_UserAccessModeType eUserMode )
```

Modify the access mode for a master to a specific region.

This function is Reentrant

Parameters

in	<i>u8RegionNum</i>	: Region to be modified
in	<i>eSupervisorMode</i>	: Specifies the new mode access in supervisor mode
in	<i>eUserMode</i>	: Specifies the new mode access in user mode

Module Documentation

Returns

void

Precondition

6.1.4.5 Rm_Mpu_GetErrorDetails()

```
Std_ReturnType Rm_Mpu_GetErrorDetails (
    Rm_Mpu_ErrorDetailsType * pErrorDetails )
```

Retrieve error details.

This function is Reentrant

Parameters

out	<i>pErrorDetails</i>	: Storage where the data will be saved
-----	----------------------	----------------------------------------

Returns

boolean - TRUE if an error was present, FALSE otherwise

Precondition

6.1.5 Variable Documentation

6.1.5.1 Rm_Config

```
const Rm_ConfigType Rm_Config [extern]
```

Export RM configurations.

6.2 MPU IPV Driver

6.2.1 Detailed Description

Function Reference

- void `Mpu_Ip_Init` (const `Mpu_Ip_ConfigType` *pConfig)
Initializes the Memory Protection Unit general parameters and region configurations.
- void `Mpu_Ip_SetRegionConfig` (uint8 u8RegionNum, const `Mpu_Ip_RegionConfigType` *const pUserConfigPtr)
Configures the region selected by u8RegionNum with the data from pUserConfigPtr.
- void `Mpu_Ip_Deinit` (void)
Disables the module and resets all region configurations.
- void `Mpu_Ip_EnableRegion` (uint8 u8RegionNum, boolean bEnable)
Enables or disabled a specific region.
- void `Mpu_Ip_SetAccessMode` (uint8 u8RegionNum, `Mpu_Ip_MasterType` eMaster, `Mpu_Ip_SupervisorAccessModeType` eSupervisorMode, `Mpu_Ip_UserAccessModeType` eUserMode)
Modify the access mode for a master to a specific region.
- boolean `Mpu_Ip_GetErrorDetails` (`Mpu_Ip_ErrorDetailsType` *pErrorDetails)
Retrieve error details.
- void `Mpu_Ip_Init_Privileged` (const `Mpu_Ip_ConfigType` *pConfig)
Initializes the MPU instance and memory regions configured.
- void `Mpu_Ip_SetRegionConfig_Privileged` (const uint8 regionNumber, const `Mpu_Ip_RegionConfigType` *const pUserConfigPtr)
Configures the specified region number using the input region configuration.
- void `Mpu_Ip_Deinit_Privileged` (void)
Deinitialize MPU instance.
- void `Mpu_Ip_EnableRegion_Privileged` (uint8 u8RegionNum, boolean bEnable)
Enable or disable region configuration.
- void `Mpu_Ip_SetAccessMode_Privileged` (uint8 u8RegionNum, `Mpu_Ip_MasterType` eMaster, `Mpu_Ip_SupervisorAccessModeType` eSupervisorMode, `Mpu_Ip_UserAccessModeType` eUserMode)
Modify the access mode for a master to a specific region.
- boolean `Mpu_Ip_GetErrorDetails_Privileged` (`Mpu_Ip_ErrorDetailsType` *pErrorDetails)
Retrieves error details such as address and error type.

6.2.2 Function Reference

6.2.2.1 Mpu_Ip_Init()

```
void Mpu_Ip_Init (
    const Mpu_Ip_ConfigType * pConfig )
```

Initializes the Memory Protection Unit general parameters and region configurations.

This function is non-reentrant

Module Documentation

Parameters

in	<i>pConfig</i>	pointer to configuration structure for MPU.
----	----------------	---------------------------------------------

Returns

void

Precondition

None

6.2.2.2 Mpu_Ip_SetRegionConfig()

```
void Mpu_Ip_SetRegionConfig (
    uint8 u8RegionNum,
    const Mpu_Ip_RegionConfigType *const pUserConfigPtr )
```

Configures the region selected by u8RegionNum with the data from pUserConfigPtr.

This function is Reentrant

Parameters

in	<i>u8RegionNum</i>	Region to be modified .
in	<i>pUserConfigPtr</i>	pointer to the region configuration structure for MPU.

Returns

void

Precondition

6.2.2.3 Mpu_Ip_Deinit()

```
void Mpu_Ip_Deinit (
    void )
```

Disables the module and resets all region configurations.

This function is Reentrant

Returns

Void

Precondition

None

6.2.2.4 Mpu_Ip_EnableRegion()

```
void Mpu_Ip_EnableRegion (
    uint8 u8RegionNum,
    boolean bEnable )
```

Enables or disabled a specific region.

This function is Reentrant

Parameters

in	<i>u8Region</i>	: Region to be modified
in	<i>bEnable</i>	: Specifies wheter the region is enabled or disabled

Returns

void

Precondition

None

6.2.2.5 Mpu_Ip_SetAccessMode()

```
void Mpu_Ip_SetAccessMode (
    uint8 u8RegionNum,
    Mpu_Ip_MasterType eMaster,
    Mpu_Ip_SupervisorAccessModeType eSupervisorMode,
    Mpu_Ip_UserAccessModeType eUserMode )
```

Modify the access mode for a master to a specific region.

This function is Reentrant

Parameters

in	<i>u8RegionNum</i>	: Region to be modified
in	<i>eMaster</i>	: Master to be modified
in	<i>eSupervisorMode</i>	: Specifies the new mode access in supervisor mode
in	<i>eUserMode</i>	: Specifies the new mode access in user mode

Returns

void

Precondition

None

6.2.2.6 Mpu_Ip_GetErrorDetails()

```
boolean Mpu_Ip_GetErrorDetails (  
    Mpu_Ip_ErrorDetailsType * pErrorDetails )
```

Retrieve error details.

This function is Reentrant

Parameters

out	<i>pErrorDetails</i>	: Storage where the data will be saved
-----	----------------------	----------------------------------------

Returns

boolean - TRUE if an error was present, FALSE otherwise

Precondition

None

6.2.2.7 Mpu_Ip_Init_Privileged()

```
void Mpu_Ip_Init_Privileged (  
    const Mpu_Ip_ConfigType * pConfig )
```

Initializes the MPU instance and memory regions configured.

6.2.2.8 Mpu_Ip_SetRegionConfig_Privileged()

```
void Mpu_Ip_SetRegionConfig_Privileged (
    const uint8 regionNumber,
    const Mpu_Ip_RegionConfigType *const pUserConfigPtr )
```

Configures the specified region number using the input region configuration.

6.2.2.9 Mpu_Ip_Deinit_Privileged()

```
void Mpu_Ip_Deinit_Privileged (
    void )
```

Deinitialize MPU instance.

6.2.2.10 Mpu_Ip_EnableRegion_Privileged()

```
void Mpu_Ip_EnableRegion_Privileged (
    uint8 u8RegionNum,
    boolean bEnable )
```

Enable or disable region configuration.

6.2.2.11 Mpu_Ip_SetAccessMode_Privileged()

```
void Mpu_Ip_SetAccessMode_Privileged (
    uint8 u8RegionNum,
    Mpu_Ip_MasterType eMaster,
    Mpu_Ip_SupervisorAccessModeType eSupervisorMode,
    Mpu_Ip_UserAccessModeType eUserMode )
```

Modify the access mode for a master to a specific region.

6.2.2.12 Mpu_Ip_GetErrorDetails_Privileged()

```
boolean Mpu_Ip_GetErrorDetails_Privileged (
    Mpu_Ip_ErrorDetailsType * pErrorDetails )
```

Retrieves error details such as address and error type.

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, QorIQ, QorIQ 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.

