User Manual

for S32K1_S32M24X MEM_43_EEP Driver

Document Number: UM2MEM_43_EEPASRR21-11 Rev0000R2.0.0 Rev. 1.0

Chapter 1

Revision History

Revision	Date	Author	Description
1.0	04.08.2023	NXP RTD Team	S32K1_S32M24X Real-Time Drivers AUTOSAR 4.4 & R21-11 Version 2.0.0

Chapter 2

Introduction

This User Manual describes NXP Semiconductor AUTOSAR MEM_EEP for S32K1_S32M24X. AUTOSAR M↩EM_EEP driver configuration parameters and deviations from the specification are described in Driver chapter of this document. AUTOSAR MEM_EEP driver requirements and APIs are described in the AUTOSAR MEM_EEP driver software specification 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 / MWCT1014S_lqfp64
- s32k144_lqfp100 / MWCT1014S_lqfp100
- s32k144_mapbga100
- s32k144w_lqfp48
- s32k144w_lqfp64
- s32k146_lqfp64
- s32k146_lqfp100 / MWCT1015S_lqfp100
- s32k146_mapbga100 / MWCT1015S_mapbga100
- s32k146_lqfp144
- s32k148_lqfp100
- s32k148_mapbga100 / MWCT1016S_mapbga100
- s32k148_lqfp144
- s32k148_lqfp176
- s32m241_lqfp64
- s32m242_lqfp64
- s32m243_lqfp64
- s32m244_lqfp64

All of the above microcontroller devices are collectively named as S32K1_S32M24X. Note: MWCT part numbers contain NXP confidential IP for Qi Wireless Power

2.2 Overview

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

AUTOSAR:

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

2.3 About This Manual

This Technical Reference employs the following typographical conventions:

- **Boldface** style: Used for important terms, notes and warnings.
- *Italic* style: Used for code snippets in the text. Note that C language modifiers such "const" or "volatile" are sometimes omitted to improve readability of the presented code.

Notes and warnings are shown as below:

Note

> This is a note.

Warning

> This is a warning

2.4 Acronyms and Definitions

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
DET	Default Error Tracer
DEM	Diagnostic Event Manager
ECC	Error Correcting Code
VLE	Variable Length Encoding
N/A	Not Available
MCU	Microcontroller Unit
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read-Only Memory
EA	EEPROM Abstraction
MEM_EEP	Memory EEPROM driver
XML	Extensible Markup Language

2.5 Reference List

#	Title	Version
1	Specification of MEM Driver	AUTOSAR Release R21-11
2	Reference Manual	S32K1xx Series Reference Manual, Rev. 14, 09/2021
		S32M24x Reference Manual, Rev. 2 Draft A, 05/2023
3	Errata	S32K116_0N96V Rev. 22/OCT/2021
		S32K118_0N97V Rev. 22/OCT/2021
		S32K142_0N33V Rev. 22/OCT/2021
		S32K144_0N57U Rev. 22/OCT/2021
		S32K144W_0P64A Rev. 22/OCT/2021
		S32K146_0N73V Rev. 22/OCT/2021
		S32K148_0N20V Rev. 22/OCT/2021
		S32M244_P64A+P73G, Rev. 0
		S32M242_N33V+P73G, Rev. 0, 6/2023
4	Data Sheet	S32K1xx Data Sheet, Rev. 14, 08/2021
		S32M2xx Data Sheet, Supports S32M24x and S32M27x, Rev. 3 Draft A, 05/2023

Chapter 3

Driver

3.1 Requirements

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

3.2 Driver Design Summary

- The MEM_EEP driver provides services for reading, writing and erasing FTFC memory. The MEM_EEP module shall work with FlexRam block of FTFC, it starts at address 0x14000000.
- The sizes of flash memory (FTFC) types on the chip are:

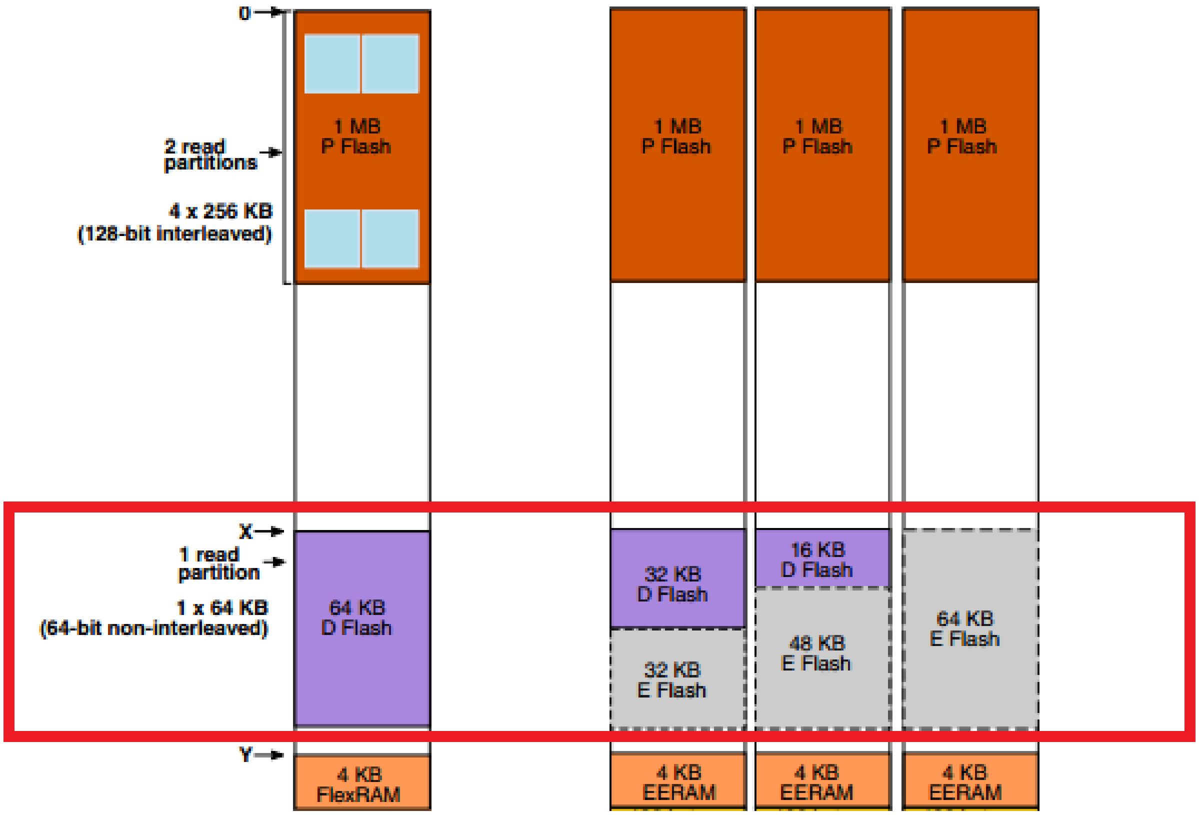
Chips	FlexRam Size
S32K116	2 KB
S32K118	2 KB
S32K142	4 KB
S32K144	4 KB
S32K144W	4 KB
S32K146	4 KB
S32K148	4 KB
S32M244	4 KB

Note

- Before working with EEP driver, we have to configure FlexNVM partition to split the FlexNVM block between data flash memory and emulated EEPROM backup memory supporting emulated EEPROM functions
- To configure FlexNVM partition, we program the DEPART value of FlexNVM partition code.
- Example for S32K148 deravative with 64KB FlexNVM read partition and 4KB FlexRAM:

DEPART	Data flash size (KB)	EEPROM backup size (KB)
0000	64	0
0011	32	32
0100	0	64
1000	0	64
1010	16	48
1011	32	32
1100	64	0
1111	Default value	

Flash memory map:



3.3 Hardware Resources

The MEM_EEP driver uses the FTFC hardware resources.

3.4 Deviations from Requirements

The driver deviates from the AUTOSAR MEM_EEP Driver software specification in some places. The table identifies the AUTOSAR requirements that are not fully implemented, implemented differently, not available, not testable or out of scope for the MEM_EEP Driver.

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

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

Requirement	Status	Description	Notes
CPR_RTD_00011.mem_eep	N/S	ISR shall check whether its respective driver is initialized. If the driver is not initialized, the ISR shall only clear interrupt status flag and return immediately.	Not applicable for MEM_EEP driver on S32K1_S32M24X
CPR_RTD_00028.mem_eep	N/S	ISR shall check whether its respective interrupt status flag and interrupt enable flag are set; i.e. whether the interrupt is not spurious. If at least one flag is not set, then the interrupt is spurious and the ISR shall only clear the interrupt status flag and return immediately.	Not applicable for MEM_EEP driver on S32K1_S32M24X
CPR_RTD_00516.mem_eep	N/S	Interrupt flags shall be cleared when the interrupt is acknowledged.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00062	N/S	If the memory hardware provides ECC information, the Mem driver shall check for correctable ECC errors and set the job result code to MEM_EC←C_CORRECTED and proceed with the current job processing.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00077	N/S	In case the last memory operation was completed but the ECC circuit corrected an ECC error, the job result shall be set to M←EM_ECC_CORRECTED.	Not applicable for MEM_EEP driver on S32K1_S32M24X

Requirement	Status	Description	Notes
SWS_Mem_00080	N/S	The service Mem_Suspend shall suspend any ongoing flash operations using an according hardware mechanism.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00083	N/S	In case a suspend operation is already in pending, Mem↩_Suspend shall reject the request by returning E_NOT_↩OK without further actions.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00085	N/S	If development error detection is enabled by MemGeneral.↩MemDevErrorDetect, the service Mem_Suspend shall check that the Mem driver has been initialized. If this check fails, Mem_Suspend shall raise the development error MEM_E_↩UNINIT.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00081	N/S	The service Mem_Resume shall resume a flash operations that was suspended by the service Mem_Suspend.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00084	N/S	In case no suspend operation is pending, Mem_Resume shall reject the request by returning E↩_NOT_OK without further actions.	Not applicable for MEM_EEP driver on S32K1_S32M24X
SWS_Mem_00086	N/S	If development error detection is enabled by MemGeneral.↩MemDevErrorDetect, the service Mem_Resume shall check that the Mem driver has been initialized. If this check fails, Mem_Resume shall raise the development error MEM_E_U↩NINIT.	Not applicable for MEM_EEP driver on S32K1_S32M24X

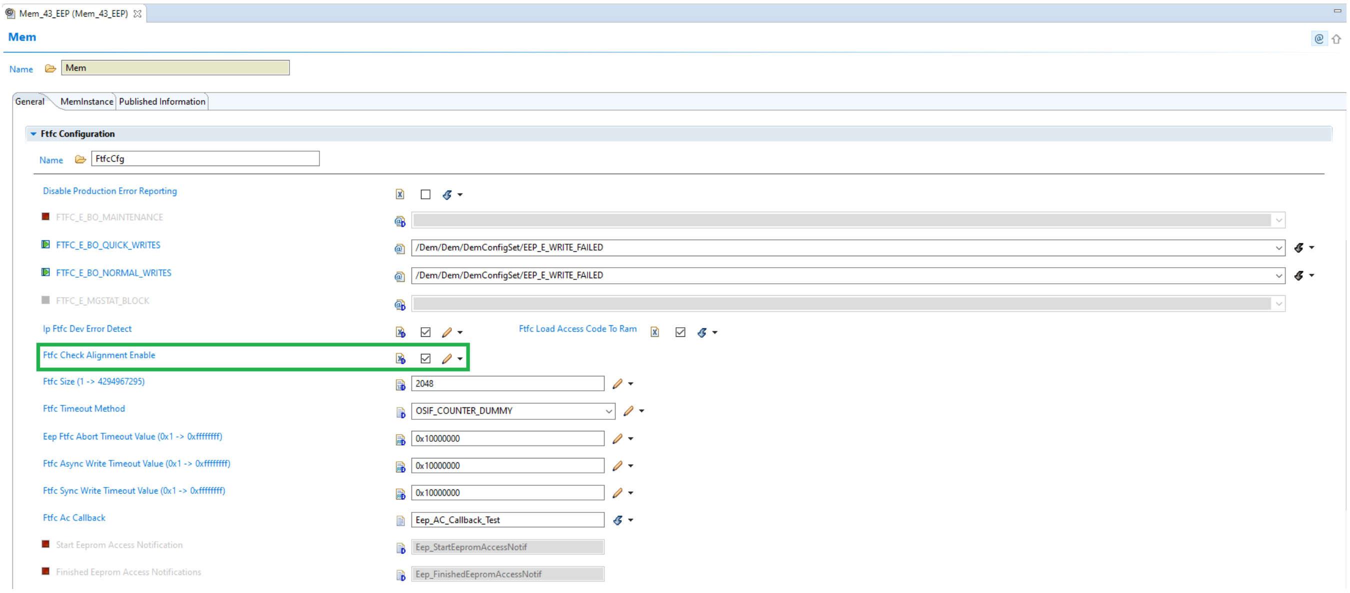
3.5 Driver Limitations

FTFC should not be run simultanously in MEM_EEP and FLS.

3.6 Driver usage and configuration tips

- Mem_43_EEP_QuickWrite now performs the first write synchronously, to leverage the driver.
- Mem_43_Eep_AcLoad and Mem_43_Eep_AcUnload are no longer used during runtime to put the AC Code in RAM, this is done at precompile time.

- Alignment handling on write function: Derivative using core M0+ does not support access unaligned address. By enable **Ftfc Check Alignment Enable**, the driver will handle unalign source address (on HLD) or check alignment (on IPL if enable dev error detect) . If that node is not enabled, users need to make sure that start address of source buffer is aligned with **Page size**. User should set parameter **Ftfc Check Alignment Enable** as FALSE for Cortex-M0 processor because there is no support for unaligned accesses.



Name: Mem

Name: FtfcCfg

Disable Production Error Reporting	☐
FTFC_E_BO_MAINTENANCE	
FTFC_E_BO_QUICK_WRITES	/Dem/Dem/DemConfigSet/EEP_E_WRITE_FAILED
FTFC_E_BO_NORMAL_WRITES	/Dem/Dem/DemConfigSet/EEP_E_WRITE_FAILED
FTFC_E_MGSTAT_BLOCK	
Ip Ftfc Dev Error Detect	☑
Ftfc Load Access Code To Ram	☑
Ftfc Check Alignment Enable	☑
Ftfc Size (1 -> 4294967295)	2048
Ftfc Timeout Method	OSIF_COUNTER_DUMMY
Eep Ftfc Abort Timeout Value (0x1 -> 0xffffffff)	0x10000000
Ftfc Async Write Timeout Value (0x1 -> 0xffffffff)	0x10000000
Ftfc Sync Write Timeout Value (0x1 -> 0xffffffff)	0x10000000
Ftfc Ac Callback	Eep_AC_Callback_Test
Start Eeprom Access Notification	Eep_StartEepromAccessNotif
Finished Eeprom Access Notifications	Eep_FinishedEepromAccessNotif

3.7 Runtime errors

The driver generates the following DEM errors at runtime.

Function	Error Code	Condition triggering the error
Eep_MainFunction()	FTFC_DEM_EVENT_BO_MAINTE↩ NANCE	Brown out detected before completing E↩ EPROM quick write maintenance (HW).
Eep_MainFunction()	FTFC_DEM_EVENT_BO_QUICK_↩ WRITE	Brown out detected before completing E↩ EPROM quick writes (HW)
Eep_MainFunction()	FTFC_DEM_EVENT_BO_NORMAL↩ _WRITE	Brown out detected before completing E↩ EPROM normal writes (HW).
Eep_MainFunction()	FTFC_DEM_EVENT_BO_MGSTAT↩ _BLOCK	Brown out detected before completing E↩ EPROM normal writes (HW).

The driver generates the following DET errors at runtime.

Function	Error Code	Condition triggering the error
MEM_43_EEP_E_UNINIT	0x01	API service called without module initialization
MEM_43_EEP_E_PARAM_POINTER	0x02	NULL_PTR passed
MEM_43_EEP_E_PARAM_ADDRESS	0x03	Address not in range

Function	Error Code	Condition triggering the error
MEM_43_EEP_E_PARAM_LENGTH	0x04	Length not in range
MEM_43_EEP_E_PARAM_INSTANCE_ID	0x05	API service called with invalid instace id
MEM_43_EEP_E_JOB_PENDING	0x06	API service called while driver still busy
MEM_43_EEP_E_TIMEOUT	0x07	The hardware operation did not finish before timeout expired.

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 Mem
  - Container MemGeneral
    - Parameter MemDevErrorDetect
    - Parameter MemEnableCheckCfgCrc
  - Container MemInstance
    - Parameter MemInstanceId
    - Container MemSectorBatch
      - Parameter MemEraseSectorSize
      - Parameter MemNumberOfSectors
      - Parameter MemReadPageSize
      - Parameter MemSpecifiedEraseCycles
      - Parameter MemStartAddress
      - Parameter MemWritePageSize
      - Container MemBurstSettings
      - Parameter MemEraseBurstSize
      - Parameter MemReadBurstSize
      - Parameter MemWriteBurstSize
  - Container MemPublishedInformation
    - Parameter MemErasedValue
  - Container FtfcCfg
    - Parameter DisableDemReportErrorStatus
    - Parameter FtfcIpDevErrorDetect
    - Parameter FtfcAcLoadToRam
    - Parameter FtfcCheckAlignmentEnabled
    - Parameter FtfcSize
    - Parameter FtfcTimeoutMethod
    - Parameter FtfcAbortTimeout
    - Parameter FtfcAsyncWriteTimeout

    - Parameter FtfcSyncWriteTimeout
    - Parameter FtfcAcCallBack
    - Parameter StartEepromAccessNotif
    - Parameter FinishedEepromAccessNotif
    - Reference FTFC_E_BO_MAINTENANCE
    - Reference FTFC_E_BO_QUICK_WRITES
    - Reference FTFC_E_BO_NORMAL_WRITES
    - Reference FTFC_E_MGSTAT_BLOCK
  - Container AutosarExt
    - Parameter MemSuspendApi
    - Parameter MemResumeApi
    - Parameter MemPropagateErrorApi
    - Parameter MemBlankCheckApi
    - Parameter MemServiceCompare
    - Parameter MemServiceQuickWriteApi
    - Parameter MemGetJobResultApi
    - Parameter MemVersionInfoApi
    - Parameter MemBlockSizeWriteAsynchBehaviorEn
    - Parameter MemBlockSizeEraseAsynchBehaviorEn
    - Parameter MemEnableUserModeSupport
  - Container MemInitConfiguration
    - Parameter MemMaxReadBlockSize
    - Parameter MemMaxWriteBlockSize
  - Container CommonPublishedInformation
    - Parameter ArReleaseMajorVersion
    - Parameter ArReleaseMinorVersion
    - Parameter ArReleaseRevisionVersion
    - Parameter ModuleId
    - Parameter SwMajorVersion
    - Parameter SwMinorVersion
    - Parameter SwPatchVersion
    - Parameter VendorApiInfix
    - Parameter VendorId

4.1 Module Mem

Configuration of the Mem_Eep (internal or external eeprom driver) module.

Included containers:

- MemGeneral
- MemInstance
- MemPublishedInformation
- FtfcCfg
- AutosarExt
- MemInitConfiguration
- CommonPublishedInformation

Property	Value
type	ECUC-MODULE-DEF
lowerMultiplicity	0
upperMultiplicity	Infinite
postBuildVariantSupport	false
supportedConfigVariants	VARIANT-PRE-COMPILE

4.2 Container MemGeneral

Container for general configuration parameters of the mem_eeprom driver. These parameters are always pre-compile.

Included subcontainers:

- None

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

4.3 Parameter MemDevErrorDetect

Pre-processor switch to enable and disable development error detection.

true : Development error detection enabled.

false: Development error detection disabled.

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

4.4 Parameter MemEnableCheckCfgCrc

Vendor specific:

true: Enable calculates CRC for Mem Eep Configuration

false: Disable calculates CRC for Mem Eep Configuration.

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
defaultValue	true

4.5 Container MemInstance

This container includes the Mem driver instance specific configuration parameters.

Included subcontainers:

- MemSectorBatch

Property	Value
type	ECUC-PARAM-CONF-CONTAINER-DEF
lowerMultiplicity	1
upperMultiplicity	65535
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE

4.6 Parameter MemInstanceId

This value specifies the unique numeric identifier which is used to reference a Mem driver instance in case multiple devices of the same type shall be addressed by one Mem driver. This value will be assigned to the symbolic name derived of the MemInstance container short name.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	0
max	65535
min	0

4.7 Container MemSectorBatch

Configuration description of a programmable sector or sector batch. Sector batch means that homogenous and coherent sectors can be configured as one MemSector element.

Included subcontainers:

- MemBurstSettings

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

4.8 Parameter MemEraseSectorSize

Size of this sector in bytes.

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

4.9 Parameter MemNumberOfSectors

Number of contiguous sectors with identical values for MemSectorSize and MemPageSize.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	65535
min	1

4.10 Parameter MemReadPageSize

Size of a read page of this sector in bytes.

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

4.11 Parameter MemSpecifiedEraseCycles

Number of erase cycles specified for the memory device (usually given in the device data sheet).

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	100000
max	4294967295
min	0

4.12 Parameter MemStartAddress

Physical start address of the sector (batch).

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

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	N/A
max	9223372036854775807
min	0

4.13 Parameter MemWritePageSize

Size of a write page of this sector in bytes.

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

4.14 Container MemBurstSettings

Container for burst setting configuration parameters of the Mem driver.

A sector burst can be used for improved performance.

Included subcontainers:

- None

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

4.15 Parameter MemEraseBurstSize

Size of sector erase burst in bytes. A sector burst can be used for

improved performance and is typically (a subset of) a sector batch.

To make use of the sector erase burst feature, the physical start

address of the sector batch must be aligned to the sector erase burst size.

Property	Value
type	ECUC-INTEGER-PARAM-DEF
origin	AUTOSAR_ECUC
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	4096
max	4294967295
min	1

4.16 Parameter MemReadBurstSize

This value specifies the maximum number of bytes the MemAcc

module requests within one Mem read request.

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

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	1
max	4294967295
min	1

4.17 Parameter MemWriteBurstSize

Size of page write/program burst in bytes. A sector burst can be used

for improved performance and is typically (a subset of) a sector batch.

To make use of the write burst feature, the physical start address must

be aligned to the write burst size.

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

4.18 Container MemPublishedInformation

Additional published parameters not covered by CommonPublishedInformation container.
Note that these parameters do not have any configuration class setting, since they are published information.
Included subcontainers:

- None

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

4.19 Parameter MemErasedValue

The contents of an erased eeprom memory cell.

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

4.20 Container FtfcCfg

Configuration for ftfc ip layer

Included subcontainers:

- None

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

4.21 Parameter DisableDemReportErrorStatus

Vendor specific: Switches the Diagnostic Error Reporting and Notification OFF.

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
defaultValue	true

4.22 Parameter FtfcIpDevErrorDetect

Pre-processor switch to enable and disable development error detection for IP layer.

true: Development error detection enabled for Ip layer.

false: Development error detection disabled for Ip layer.

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
defaultValue	true

4.23 Parameter FtfcAcLoadToRam

The eeprom driver shall load the eeprom access code to RAM.

true: Eeprom access code will be loaded to RAM.

false: Eeprom access code will be stored in ROM.

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
defaultValue	false

4.24 Parameter FtfcCheckAlignmentEnabled

Compile switch to enable and disable checking the RAM addresses need to be 2bytes or 4bytes aligned when performing 2bytes or 4bytes reads/writes.

true: Enable checking.

false: Disable checking.

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
defaultValue	false

4.25 Parameter FtfcSize

Ftfc Size (in byte).

It depend on size of Eep backup region.

Ex: If size of Eep backup region is 32kb, this value must be between 1 and 2048(32/16=2kb).

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
defaultValue	2048
max	4294967295
min	1

4.26 Parameter FtfcTimeoutMethod

Vendor specific: Counter type used in timeout detection for Ftfc operations.

Based on selected counter type the timeout value will be interpreted as follows:

OSIF_COUNTER_SYSTEM - Microseconds.

OSIF_COUNTER_CUSTOM - Defined by user implementation of timing services

OSIF_COUNTER_DUMMY - Counts the number of iterations of the waiting loop. The actual timeout depends on many factors: operation type, compiler optimizations, interrupts or other tasks in the system, etc.

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
defaultValue	OSIF_COUNTER_DUMMY
literals	['OSIF_COUNTER_SYSTEM', 'OSIF_COUNTER_CUSTOM', 'OSIF_CO↩ UNTER_DUMMY']

4.27 Parameter FtfcAbortTimeout

Ftfc Abort Timeout Value

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
defaultValue	268435456
max	4294967295
min	1

4.28 Parameter FtfcAsyncWriteTimeout

Ftfc Async Write Timeout Value

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
defaultValue	268435456
max	4294967295
min	1

4.29 Parameter FtfcSyncWriteTimeout

Ftfc Sync Write Timeout Value

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
defaultValue	268435456
max	4294967295
min	1

4.30 Parameter FtfcAcCallBack

Vendor specific: Mapped to the Access Code Callback provided by some upper layer module.

Note: Disable the Access Code Callback to have it set as NULL_PTR.

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
defaultValue	NULL_PTR

4.31 Parameter StartEepromAccessNotif

Start eeprom access. If configured, this notification will be called before any EFLASH memory access in synchronuous mode.

The purpose of this notification together with EepFinishedEepromAccess,

is to be used by the integrator in case operations are needed before and after EFLASH access, for eg to avoid concurrent access issue.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	Eep_StartEepromAccessNotif

4.32 Parameter FinishedEepromAccessNotif

Finished eeprom access. If configured, this notification will be called before any EFLASH memory access in synchronuous mode.

The purpose of this notification together with EepStartEepromAccess,

is to be used by the integrator in case operations are needed before and after EFLASH access, for eg to avoid concurrent access issue.

Property	Value
type	ECUC-FUNCTION-NAME-DEF
origin	NXP
symbolicNameValue	false
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	Eep_FinishedEepromAccessNotif

4.33 Reference FTFC_E_BO_MAINTENANCE

Reference to the DemEventParameter which shall be issued when the error "Brown out detected before completing EEPROM quick write maintenance (HW)" has occurred.

Property	Value
type	ECUC-REFERENCE-DEF

Property	Value
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.34 Reference FTFC_E_BO_QUICK_WRITES

Reference to the DemEventParameter which shall be issued when the error "Brown out detected before completing EEPROM quick writes (HW)" has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.35 Reference FTFC_E_BO_NORMAL_WRITES

Reference to the DemEventParameter which shall be issued when the error "Brown out detected before completing EEPROM normal writes (HW)" has occurred.

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

Property	Value
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.36 Reference FTFC_E_MGSTAT_BLOCK

Reference to the DemEventParameter which shall be issued when the error "Brown out detected before completing EEPROM normal writes (HW)" has occurred.

Property	Value
type	ECUC-REFERENCE-DEF
origin	NXP
lowerMultiplicity	0
upperMultiplicity	1
postBuildVariantMultiplicity	false
multiplicityConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
requiresSymbolicNameValue	False
destination	/AUTOSAR/EcucDefs/Dem/DemConfigSet/DemEventParameter

4.37 Container AutosarExt

Vendor specific: This container contains the global Non-Autosar configuration parameters of the Mem_Eep driver.

This container is a MultipleConfigurationContainer, i.e. this container and

its sub-containers exist once per configuration set.

Included subcontainers:

- None

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

4.38 Parameter MemSuspendApi

Compile switch to enable and disable the Mem_Suspend function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	true

4.39 Parameter MemResumeApi

Compile switch to enable and disable the Mem_Resume function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	true

4.40 Parameter MemPropagateErrorApi

Compile switch to enable and disable the Mem_PropageError function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	true

4.41 Parameter MemBlankCheckApi

Compile switch to enable and disable the Mem_BlankCheck function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	true

4.42 Parameter MemServiceCompare

Compile switch to enable and disable the MemServiceCompare feature.

true: feature supported / feature provided.

false: feature not supported / feature not provided

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
defaultValue	true

4.43 Parameter MemServiceQuickWriteApi

Compile switch to enable and disable the Eep_EnableQuickWrites function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	false

4.44 Parameter MemGetJobResultApi

Compile switch to enable and disable the Mem_GetJobResult function.

true: API supported / function provided.

false: API not supported / function not provided

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
defaultValue	true

4.45 Parameter MemVersionInfoApi

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

true: Version info API enabled.

false: Version info API 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
defaultValue	true

4.46 Parameter MemBlockSizeWriteAsynchBehaviorEn

Vendor specific: Enable asynchronous execution of the write job in the Mem_MainFunction function which doesn't wait (block) for completion of the block size write operation(s).

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
defaultValue	false

4.47 Parameter MemBlockSizeEraseAsynchBehaviorEn

Vendor specific: Enable asynchronous execution of the erase job in the Mem_MainFunction function which doesn't wait (block) for completion of the block size erase operation(s).

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
defaultValue	false

4.48 Parameter MemEnableUserModeSupport

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

configuring REG_PROT for Mem IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1

for more information and availability on this platform, please see chapter User Mode Support in IM

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
defaultValue	false

4.49 Container MemInitConfiguration

Container for runtime configuration parameters of the eeprom driver.

Implementation Type: Mem_EEP_ConfigType.

Included subcontainers:

- None

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

4.50 Parameter MemMaxReadBlockSize

The maximum number of bytes to read in one cycle of the eeprom driver's job processing function.

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

Property	Value
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PRE-COMPILE
defaultValue	4096
max	4294967295
min	0

4.51 Parameter MemMaxWriteBlockSize

The maximum number of bytes to write in one cycle of the eeprom driver's job processing function.

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
defaultValue	4096
max	4294967295
min	0

4.52 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

4.53 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
defaultValue	4
max	4
min	4

4.54 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
defaultValue	7
max	7
min	7

4.55 Parameter ArReleaseRevisionVersion

Vendor specific: Patch 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
defaultValue	0
max	0
min	0

4.56 Parameter ModuleId

Vendor specific: Module ID of this module.

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
defaultValue	90
max	90
min	90

4.57 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

Property	Value
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	2
max	2
min	2

4.58 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: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.59 Parameter SwPatchVersion

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

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

Property	Value
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	0
max	0
min	0

4.60 Parameter VendorApiInfix

In driver modules which can be instantiated several times on a single ECU, BSW00347 requires that the name of APIs is extended by the VendorId and a vendor specific name.

This parameter is used to specify the vendor specific name. In total, the implementation specific name is generated as follows:

<ModuleName>_>VendorId>_<VendorApiInfix>.

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

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

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

4.61 Parameter VendorId

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

Property	Value
type	ECUC-INTEGER-PARAM-DEF

Property	Value
origin	NXP
symbolicNameValue	false
lowerMultiplicity	1
upperMultiplicity	1
postBuildVariantMultiplicity	N/A
multiplicityConfigClasses	N/A
postBuildVariantValue	false
valueConfigClasses	VARIANT-PRE-COMPILE: PUBLISHED-INFORMATION
defaultValue	43
max	43
min	43

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

Chapter 5

Module Index

5.1 Software Specification

Here is a list of all modules:

Chapter 6

Module Documentation

6.1 FTFC_MEM_EEP_IP

6.1.1 Detailed Description

Function Reference

- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Init (Ftfc_Mem_Eep_Ip_ConfigType const *p↩ Config)

  Initialize the module.
- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Read (Ftfc_Mem_Eep_Ip_AddressType SrcAddress, uint8 *pu8DestAddress, Ftfc_Mem_Eep_Ip_LengthType Length)

  Read `Length` bytes from EEPROM `SrcAddress` to `pu8DestAddress`.
- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Compare (Ftfc_Mem_Eep_Ip_AddressType Src↩ Address, uint8 const *pu8DestAddress, Ftfc_Mem_Eep_Ip_LengthType Length)

  Compare the first `Length` bytes of `pu8DestAddress` to the contents found at `SrcAddress`.
- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Write (Ftfc_Mem_Eep_Ip_AddressType Dest↩ Address, uint8 const *pu8SrcAddress, Ftfc_Mem_Eep_Ip_PageSizeType PageSize, boolean Async)

  Write `PageSize` bytes from `pu8SrcAddress` buffer to EEPROM at `offset` DestAddress.
- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Erase (Ftfc_Mem_Eep_Ip_AddressType Address, Ftfc_Mem_Eep_Ip_PageSizeType PageSize, boolean Async)

  Erase API.
- Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_GetJobResult (void)

  Interrogate the result of the last async job, considering the timeout and FSTAT errors.
- Ftfc_Mem_Eep_Ip_BrownOutCodeType Ftfc_Mem_Eep_Ip_GetBrownOutCode (void)

  Getter for Ftfc_Mem_Eep_Ip_eBrownOutCode.

6.1.2 Function Reference

6.1.2.1 Ftfc_Mem_Eep_Ip_Init()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Init (
            Ftfc_Mem_Eep_Ip_ConfigType const * pConfig )
```

Initialize the module.

Set FlexRAM to work as EEERAM. Query EEPROM quick write status and complete the maintenance, if needed.

Out of reset with the FSTAT[CCIF] bit clear, the partition settings (EEESIZE, DEPART) are read from the data flash IFR and the emulated EEPROM file system is initialized accordingly. The emulated EEPROM file system locates all valid EEPROM data records in EEPROM backup and copies the newest data to FlexRAM.

CCIF is cleared throughout the reset sequence. Completion of the reset sequence is marked by setting CCIF which enables flash user commands.

Parameters

in	*pConfig*	pointer stored in Ftfc_Mem_Eep_Ip_pxConfiguration

Returns

the initialization result

Return values

FTFC_MEM_EEP_IP_STATUS_OK	initialization successful
FTFC_MEM_EEP_IP_STATUS_TIMEOUT	a flash cmd timeout has occurred
FTFC_MEM_EEP_IP_STATUS_FAILED	a flash command failed to execute
FTFC_MEM_EEP_IP_STATUS_FAILED_MGSTAT	one or more MGSTAT 1/2/3 bits were set

6.1.2.2 Ftfc_Mem_Eep_Ip_Read()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Read (
            Ftfc_Mem_Eep_Ip_AddressType SrcAddress,
            uint8 * pu8DestAddress,
            Ftfc_Mem_Eep_Ip_LengthType Length )
```

Read `Length` bytes from EEPROM `SrcAddress` to `pu8DestAddress`.

Parameters

in	*SrcAddress*	EEPROM address to read from
out	*pu8DestAddress*	buffer to store the read data
in	*Length*	how many bytes to read

Returns

the read operation status

Return values

FTFC_MEM_EEP_IP_STATUS_OK	the requested bytes were copied into the destination buffer
FTFC_MEM_EEP_IP_STATUS_FAILED	FTFC not ready
FTFC_MEM_EEP_IP_STATUS_FAILED	a read was attempted on an invalid page size

6.1.2.3 Ftfc_Mem_Eep_Ip_Compare()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Compare (
             Ftfc_Mem_Eep_Ip_AddressType SrcAddress,
             uint8 const * pu8DestAddress,
             Ftfc_Mem_Eep_Ip_LengthType Length )
```

Compare the first `Length` bytes of `pu8DestAddress` to the contents found at `SrcAddress`.

Parameters

in	*pu8DestAddress*	pointer to the data buffer
in	*SrcAddress*	where the contents in EEPROM are stored
in	*Length*	how many bytes to compare

Returns

the comparison result

Return values

FTFC_MEM_EEP_IP_STATUS_OK	the contents match
FTFC_MEM_EEP_IP_STATUS_BLOCK_INCONSISTENT	the contents do not match
FTFC_MEM_EEP_IP_STATUS_FAILED	FTFC not ready
FTFC_MEM_EEP_IP_STATUS_FAILED	a read was attempted on an invalid page size

6.1.2.4 Ftfc_Mem_Eep_Ip_Write()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Write (
             Ftfc_Mem_Eep_Ip_AddressType DestAddress,
             uint8 const * pu8SrcAddress,
             Ftfc_Mem_Eep_Ip_PageSizeType PageSize,
             boolean Async )
```

Write `PageSize` bytes from `pu8SrcAddress` buffer to EEPROM at `offset` DestAddress.

Parameters

out	*DestAddress*	EEPROM offset
in	*pu8SrcAddress*	buffer containing the data to be written
in	*PageSize*	must be a valid PageSize: 1, 2 or 4 bytes for FTFC and only 4 bytes for FTFM
in	*Async*	choose between a synchronous and an asynchronous job

Returns

the write operation result

Return values

FTFC_MEM_EEP_IP_STATUS_FAILED	FTFC not ready
FTFC_MEM_EEP_IP_STATUS_FAILED	an invalid page alignment was given
FTFC_MEM_EEP_IP_STATUS_FAILED	sync: some of the FSTAT error bits were set
FTFC_MEM_EEP_IP_STATUS_TIMEOUT	sync: timeout occurred while waiting for CCIF
FTFC_MEM_EEP_IP_STATUS_OK	sync: page successfully written to EFLASH
FTFC_MEM_EEP_IP_STATUS_PENDING	async: the page was written to FlexRAM, but the status of the EFLASH record shall be interrogated with GetJobResult

6.1.2.5 Ftfc_Mem_Eep_Ip_Erase()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_Erase (
            Ftfc_Mem_Eep_Ip_AddressType Address,
            Ftfc_Mem_Eep_Ip_PageSizeType PageSize,
            boolean Async )
```

Erase API.

Invokes a write with ERASED_CELL_VALUE.

Returns

the erase operation result

Return values

FTFC_MEM_EEP_IP_STATUS_FAILED	FTFC not ready
FTFC_MEM_EEP_IP_STATUS_FAILED	an invalid page size/alignment was given
FTFC_MEM_EEP_IP_STATUS_FAILED	sync: some of the FSTAT error bits were set
FTFC_MEM_EEP_IP_STATUS_TIMEOUT	sync: timeout occurred while waiting for CCIF
FTFC_MEM_EEP_IP_STATUS_OK	sync: page successfully written to EFLASH
FTFC_MEM_EEP_IP_STATUS_PENDING	async: the page was written to FlexRAM, but the status of the EFLASH record shall be interrogated with GetJobResult

6.1.2.6 Ftfc_Mem_Eep_Ip_GetJobResult()

```
Ftfc_Mem_Eep_Ip_StatusType Ftfc_Mem_Eep_Ip_GetJobResult (
            void  )
```

Interrogate the result of the last async job, considering the timeout and FSTAT errors.

Returns

the result of the last async job

Return values

FTFC_MEM_EEP_IP_STATUS_OK	the job finished successfully
FTFC_MEM_EEP_IP_STATUS_FAILED	FSTAT error bits were set
FTFC_MEM_EEP_IP_STATUS_PENDING	the job is still waiting for CCIF
FTFC_MEM_EEP_IP_STATUS_TIMEOUT	a timeout has occurred while waiting for CCIF

6.1.2.7 Ftfc_Mem_Eep_Ip_GetBrownOutCode()

```
Ftfc_Mem_Eep_Ip_BrownOutCodeType Ftfc_Mem_Eep_Ip_GetBrownOutCode (
            void  )
```

Getter for Ftfc_Mem_Eep_Ip_eBrownOutCode.

Returns

the brownout code read after reset

Return values

0x04	normal write was interrupted
0x02	quick write was interrupted before writing all bytes to flash
0x01	quick write was interrupted before maintenance completed

6.2 Mem_43_EEProm Driver

6.2.1 Detailed Description

Data Structures

- struct Mem_43_EEP_JobRuntimeInfoType

  Mem job runtime information Type. More...
- struct Mem_43_EEP_SectorBatchType

  Sector Batch Type. More...
- struct Mem_43_EEP_MemInstanceType

  Mem Instance Type. More...
- struct Mem_43_EEP_ConfigType

  Mem_43_EEP Config Type. More...
- struct Mem_43_EEP_CompareConfigType

  Mem_43_EEP Compare Configuration Type. More...
- struct Mem_43_EEP_QuickWriteConfigType

  Mem_43_EEP Quick Write Type. More...

Types Reference

- typedef MEM_43_EEP_ADDRESSTYPE Mem_43_EEP_AddressType

  Mem_43_EEP Address Type.
- typedef uint8 Mem_43_EEP_DataType

  Mem_43_EEP Data Type.
- typedef uint32 Mem_43_EEP_InstanceIdType

  Mem_43_EEP InstanceId Type.
- typedef MEM_43_EEP_LENGTHTYPE Mem_43_EEP_LengthType

  Mem_43_EEP Length Type.
- typedef uint32 Mem_43_EEP_HwServiceIdType

  Mem_43_EEP HwServiceId Type.

Enum Reference

- enum Mem_43_EEP_ReturnType

  Result of low-level eeprom operation.
- enum Mem_43_EEP_JobType

  Type of job currently executed by Mem_43_EEP_MainFunction.
- enum Mem_43_EEP_BlankCheckType

  Asynchronous job result type.
- enum Mem_43_EEP_CompareCheckType
- enum Mem_43_EEP_CrcDataSizeType

  Size of data to be processeed by CRC.
- enum Mem_43_EEP_JobResultType

  Asynchronous job result type.

6.2.2 Data Structure Documentation

6.2.2.1 struct Mem_43_EEP_JobRuntimeInfoType

Mem job runtime information Type.

This structure contains runtime information the current processing job of each Mem instance.

Definition at line 146 of file Mem_43_EEP_InternalTypes.h.

Data Fields

Type	Name	Description
Mem_43_EEP_LengthType	Mem_43_EEP_u32MaxRead	Maximum number of bytes to read in one cycle of Mem_43_EEP_MainFunction.
Mem_43_EEP_LengthType	Mem_43_EEP_u32MaxWrite	Maximum number of bytes to write in one cycle of Mem_43_EEP_MainFunction.
Mem_43_EEP_AddressType	Mem_43_EEP_u32EepromAddrIt	Logical address of data block currently processed by Mem_43_EEP_MainFunction.
Mem_43_EEP_LengthType	Mem_43_EEP_u32RemainingLength	Remainin length to be transfered until the end of the job.
Mem_43_EEP_JobResultType	Mem_43_EEP_eJobResult	Result of last mem_43_eeprom module job.
Mem_43_EEP_JobType	Mem_43_EEP_eJob	Type of currently executed job (erase, write, read, or blank)
Mem_43_EEP_HwServiceIdType	Mem_43_EEP_u32HwServiceIdJob	Type of currently executed HwServiceId job (compare) when executing ProcessHwSpecificServiceJob.
const Mem_43_EEP_DataType *	Mem_43_EEP_pJobSrcAddrPtr	Pointer to current position in source data buffer.
Mem_43_EEP_DataType *	Mem_43_EEP_pJobDataDestPtr	Pointer to current position in target data buffer.

6.2.2.2 struct Mem_43_EEP_SectorBatchType

Sector Batch Type.

Sector Batch data structure for group of identical sectors Note: burst sizes equal to normal sizes in case burst disabled

Definition at line 207 of file Mem_43_EEP_Types.h.

6.2.2.3 struct Mem_43_EEP_MemInstanceType

Mem Instance Type.

Mem Instance data structure

Definition at line 224 of file Mem_43_EEP_Types.h.

6.2.2.4 struct Mem_43_EEP_ConfigType

Mem_43_EEP Config Type.

Mem_43_EEP module initialization data structure

Definition at line 239 of file Mem_43_EEP_Types.h.

Data Fields

- const Mem_43_EEP_MemInstanceType ∗ MemInstances

  Point to first element in array of mem instances configurations.
- Ftfc_Mem_Eep_Ip_ConfigType const ∗ pxFtfcIpConfig

  FTFC IP Config Set.

6.2.2.4.1 Field Documentation

6.2.2.4.1.1 MemInstances `const Mem_43_EEP_MemInstanceType∗ MemInstances`

Point to first element in array of mem instances configurations.

Definition at line 244 of file Mem_43_EEP_Types.h.

6.2.2.4.1.2 pxFtfcIpConfig `Ftfc_Mem_Eep_Ip_ConfigType const∗ pxFtfcIpConfig`

FTFC IP Config Set.

Definition at line 252 of file Mem_43_EEP_Types.h.

6.2.2.5 struct Mem_43_EEP_CompareConfigType

Mem_43_EEP Compare Configuration Type.

```
Mem_43_EEP_CompareConfigType
```

Definition at line 260 of file Mem_43_EEP_Types.h.

6.2.2.6 struct Mem_43_EEP_QuickWriteConfigType

Mem_43_EEP Quick Write Type.

Definition at line 272 of file Mem_43_EEP_Types.h.

6.2.3 Types Reference

6.2.3.1 Mem_43_EEP_AddressType

```
typedef MEM_43_EEP_ADDRESSTYPE Mem_43_EEP_AddressType
```

Mem_43_EEP Address Type.

Physical memory device address type.

Definition at line 133 of file Mem_43_EEP_Types.h.

6.2.3.2 Mem_43_EEP_DataType

```
typedef uint8 Mem_43_EEP_DataType
```

Mem_43_EEP Data Type.

Read data user buffer type.

Definition at line 141 of file Mem_43_EEP_Types.h.

6.2.3.3 Mem_43_EEP_InstanceIdType

```
typedef uint32 Mem_43_EEP_InstanceIdType
```

Mem_43_EEP InstanceId Type.

Job end notification function called by Mem in case the job processing is configured for job end notification.

Definition at line 150 of file Mem_43_EEP_Types.h.

6.2.3.4 Mem_43_EEP_LengthType

```
typedef MEM_43_EEP_LENGTHTYPE Mem_43_EEP_LengthType
```

Mem_43_EEP Length Type.

Number of bytes to read,write,erase,compare

Definition at line 190 of file Mem_43_EEP_Types.h.

6.2.3.5 Mem_43_EEP_HwServiceIdType

```
typedef uint32 Mem_43_EEP_HwServiceIdType
```

Mem_43_EEP HwServiceId Type.

Job end notification function called by Mem in case the job processing is configured for job end notification.

Definition at line 199 of file Mem_43_EEP_Types.h.

6.2.4 Enum Reference

6.2.4.1 Mem_43_EEP_ReturnType

```
enum Mem_43_EEP_ReturnType
```

Result of low-level eeprom operation.

Enumerator

MEM_43_EEP_E_OK	operation succeeded
MEM_43_EEP_E_FAILED	operation failed due to hardware error
MEM_43_EEP_E_PENDING	operation is pending

Definition at line 66 of file Mem_43_EEP_InternalTypes.h.

6.2.4.2 Mem_43_EEP_JobType

```
enum Mem_43_EEP_JobType
```

Type of job currently executed by Mem_43_EEP_MainFunction.

Enumerator

MEM_43_EEP_JOB_ERASE	erase one or more complete mem_eeprom sectors
MEM_43_EEP_JOB_WRITE	write one or more complete mem_eeprom pages
MEM_43_EEP_JOB_READ	read one or more bytes from mem_eeprom memory
MEM_43_EEP_JOB_BLANKCHECK	check blank bytes from mem_eeprom memory
MEM_43_EEP_JOB_HWSPECIFICSERVICE	hardware specific service

Definition at line 76 of file Mem_43_EEP_InternalTypes.h.

6.2.4.3 Mem_43_EEP_BlankCheckType

```
enum Mem_43_EEP_BlankCheckType
```

Asynchronous job result type.

Enumerator

MEM_43_EEP_BLANKCHECK_E_OK	Has been finished successfully.
MEM_43_EEP_BLANKCHECK_E_NOT_OK	Has not been finished successfully.
MEM_43_EEP_BLANKCHECK_E_INCONSISTENT	The checked page is not blank.

Definition at line 103 of file Mem_43_EEP_InternalTypes.h.

6.2.4.4 Mem_43_EEP_CompareCheckType

```
enum Mem_43_EEP_CompareCheckType
```

Result of compare operation.

Definition at line 122 of file Mem_43_EEP_InternalTypes.h.

6.2.4.5 Mem_43_EEP_CrcDataSizeType

```
enum Mem_43_EEP_CrcDataSizeType
```

Size of data to be processeed by CRC.

Definition at line 134 of file Mem_43_EEP_InternalTypes.h.

6.2.4.6 Mem_43_EEP_JobResultType

```
enum Mem_43_EEP_JobResultType
```

Asynchronous job result type.

Enumerator

MEM_43_EEP_JOB_OK	The last job has been finished successfully.
MEM_43_EEP_JOB_PENDING	A job is currently being processed.
MEM_43_EEP_JOB_FAILED	Job failed for some unspecific reason.
MEM_43_EEP_INCONSISTENT	The checked page is not blank.
MEM_43_EEP_ECC_UNCORRECTED	Uncorrectable ECC errors occurred during memory access.
MEM_43_EEP_ECC_CORRECTED	Correctable ECC errors occurred during memory access.

Definition at line 156 of file Mem_43_EEP_Types.h.

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.