MCAL Configuration Verification Manual for MCALLIB

32-bit TriCore™ AURIX™ TC3xx microcontroller family

About this document

Scope and purpose

This Configuration Data Reference document is applicable to all TC3xx devices in the TriCore™ AURIX™ family of 32-bit microcontrollers.

The purpose of this document is to facilitate the integrator to verify the generated code based on the input configuration parameters. This document describes details of structures, defines, macros and variables generated from the configuration parameters.

Intended audience

This document is intended for integrators who need to understand the logic of the generated configuration code of AURIX™ AUTOSAR.

Reference documents

This document should be read in conjunction with the following documents:

- AURIX™ TC3xx MCAL User Manual MCALLIB

Table of contents

McalLib driver

This chapter describes the details of the configuration data generated from the MCALLIB driver.

1.1 File: McalLib_Cfg.h

The generated header file contains all pre-compile configuration parameters. Pre-compile time configuration allows decoupling of the static configuration from implementation. The file is generated in ‘inc’ folder.

1.1.1 Macro: MCALLIB_AR_RELEASE_MAJOR_VERSION

Table 1 MCALLIB_AR_RELEASE_MAJOR_VERSION

Name	MCALLIB_AR_RELEASE_MAJOR_VERSION	
Description	Major version number of AUTOSAR release on which the McalLib implementation is based on.	
Verification method	The macro is generated with the value present in ‘CommonPublishedInformation/ArMajorVersion’. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with ArMajorVersion 4	`#define MCALLIB_AR_RELEASE_MAJOR_VERSION (4U)`

1.1.2 Macro: MCALLIB_AR_RELEASE_MINOR_VERSION

Table 2 MCALLIB_AR_RELEASE_MINOR_VERSION

Name	MCALLIB_AR_RELEASE_MINOR_VERSION	
Description	Minor version number of AUTOSAR release on which the McalLib implementation is based on.	
Verification method	The macro is generated with the value present in ‘CommonPublishedInformation/ArMinorVersion’. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with ArMinorVersion 2	`#define MCALLIB_AR_RELEASE_MINOR_VERSION (2U)`

1.1.3 Macro: MCALLIB_AR_RELEASE_REVISION_VERSION

Table 3 MCALLIB_AR_RELEASE_REVISION_VERSION

Name	MCALLIB_AR_RELEASE_REVISION_VERSION
Description	Revision version number of AUTOSAR release on which the McalLib implementation is

	based on.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/ArPatchVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with ArPatchVersion 2	`#define MCALLIB_AR_RELEASE_REVISION_VERSION (4U)`

1.1.4 Macro: MCALLIB_SW_MAJOR_VERSION

Table 4 MCALLIB_SW_MAJOR_VERSION

Name	MCALLIB_SW_MAJOR_VERSION	
Description	Major version number of the McalLib module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMajorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with SwMajorVersion 10	`#define MCALLIB_SW_MAJOR_VERSION (10U)`

1.1.5 Macro: MCALLIB_SW_MINOR_VERSION

Table 5 MCALLIB_SW_MINOR_VERSION

Name	MCALLIB_SW_MINOR_VERSION	
Description	Minor version number of the McalLib module.	
Verification method	The macro is generated with the value present in 'CommonPublishedInformation/SwMinorVersion'. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with SwMinorVersion 10	`#define MCALLIB_SW_MINOR_VERSION (10U)`

1.1.6 Macro: MCALLIB_SW_PATCH_VERSION

Table 6 MCALLIB_SW_PATCH_VERSION

Name	MCALLIB_SW_PATCH_VERSION
Description	Patch level version number of the McalLib module.
Verification method	The macro is generated with the value present in

	‘CommonPublishedInformation/SwPatchVersion’. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with SwPatchVersion 0	`#define MCALLIB_SW_PATCH_VERSION (0U)`

1.1.7 Macro: MCALLIB_SAFETY_ENABLE

Table 7 MCALLIB_SAFETY_ENABLE

Name	MCALLIB_SAFETY_ENABLE	
Description	Enables/disables safety features	
Verification method	The macro is generated as STD_ON if McalLibSafetyEnable configuration parameter is set to ‘True’ else the macro is generated as STD_OFF.	
Example(s)	**Action**	**Generated output**
	McalLibSafetyEnable = True	`#define MCALLIB_SAFETY_ENABLE (STD_ON)`
	McalLibSafetyEnable = False	`#define MCALLIB_SAFETY_ENABLE (STD_OFF)`

1.1.8 Macro: MCAL_NO_OF_CORES

Table 8 MCAL_NO_OF_CORES

Name	MCAL_NO_OF_CORES	
Description	Specifies the number of cores available for the selected device.	
Verification method	The macro is generated based on number of cores available in the device. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file and total core in the device = 4	`#define MCAL_NO_OF_CORES (4)`
	Generate McalLib_Cfg.h file and total core in the device = 6	`#define MCAL_NO_OF_CORES (6)`

1.1.9 Macro: MCAL_BACKUP_FREQUENCY

Table 9 MCAL_BACKUP_FREQUENCY

Name	MCAL_BACKUP_FREQUENCY
Description	Specifies the frequency of the back-up clock in MHz.
Verification method	The macro is generated based on back-up clock available in the device.

	Note: The macro is not user configurable.	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file and Backup Frequency is 100MHz	`#define MCAL_BACKUP_FREQUENCY (100)`

1.1.10 Macro: MCAL_PSPR<x>_GLOBAL_BASE_ADDR

Table 10 MCAL_PSPR<x>_GLOBAL_BASE_ADDR

Name	MCAL_PSPR<x>_GLOBAL_BASE_ADDR	
Description	Specifies the start address of PSPR for core<x> where x=0 up to maximum number of cores available in hardware.	
Verification method	The macro is generated based on start address of PSPR for core<x> in the device. If the PSPR for the core is not available, the macro is generated as 0. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate McalLib_Cfg.h file with
	• PSPR0 start address is 1880096768
	• PSPR1 start address is 1611661312
	• PSPR2 is not available
	• PSPR3 start address is 1074790400
	• PSPR4 start address is 806354944
	• PSPR5 is not available
		`#define MCAL_PSPR0_GLOBAL_BASE_ADDR (1880096768)`
		`#define MCAL_PSPR1_GLOBAL_BASE_ADDR (1611661312)`
		`#define MCAL_PSPR2_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_PSPR3_GLOBAL_BASE_ADDR (1074790400)`
		`#define MCAL_PSPR4_GLOBAL_BASE_ADDR (806354944)`
		`#define MCAL_PSPR5_GLOBAL_BASE_ADDR (0)`

	• PSPR0 start address is 1611661312
	• PSPR1 is not available
	• PSPR2 start address is 1343225856
	• PSPR3 is not available
	• PSPR4 is not available.
	• PSPR5 start address is 269484032
		`#define MCAL_PSPR0_GLOBAL_BASE_ADDR (1611661312)`
		`#define MCAL_PSPR1_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_PSPR2_GLOBAL_BASE_ADDR (1343225856)`
		`#define MCAL_PSPR3_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_PSPR4_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_PSPR5_GLOBAL_BASE_ADDR (269484032)`

1.1.11 Macro: MCAL_PSPR<x>_GLOBAL_END_ADDR

Table 11 **MCAL_PSPR<x>_GLOBAL_END_ADDR**

Name	MCAL_PSPR<x>_GLOBAL_END_ADDR	
Description	Specifies the end address of PSPR for core<x> where x=0 up to maximum number of cores available in hardware.	
Verification method	The macro is generated based on end address of PSPR for core<x> in the device. If the PSPR for the core is not available, the macro is generated as 0. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate McalLib_Cfg.h file with
	• PSPR0 end address is 1880162303
	• PSPR1 end address is 1611726847
	• PSPR2 end address is 1343291391
	• PSPR3 is not available
	• PSPR4 is not available
	• PSPR5 end address is 269549567
		`#define MCAL_PSPR0_GLOBAL_END_ADDR (1880162303)`
		`#define MCAL_PSPR1_GLOBAL_END_ADDR (1611726847)`
		`#define MCAL_PSPR2_GLOBAL_END_ADDR (1343291391)`
		`#define MCAL_PSPR3_GLOBAL_END_ADDR (0)`
		`#define MCAL_PSPR4_GLOBAL_END_ADDR (0)`
		`#define MCAL_PSPR5_GLOBAL_END_ADDR (269549567)`

	• PSPR0 end address is 1611726847
	• PSPR1 is not available
	• PSPR2 is not available
	• PSPR3 end address is 1074855935
	• PSPR4 end address is 806420479
	• PSPR5 is not available
		`#define MCAL_PSPR0_GLOBAL_END_ADDR (1611726847)`
		`#define MCAL_PSPR1_GLOBAL_END_ADDR (0)`
		`#define MCAL_PSPR2_GLOBAL_END_ADDR (0)`
		`#define MCAL_PSPR3_GLOBAL_END_ADDR (1074855935)`
		`#define MCAL_PSPR4_GLOBAL_END_ADDR (806420479)`
		`#define MCAL_PSPR5_GLOBAL_END_ADDR (0)`

1.1.12 Macro: MCAL_DSPR<x>_GLOBAL_BASE_ADDR

Table 12 **MCAL_DSPR<x>_GLOBAL_BASE_ADDR**

Name	MCAL_DSPR<x>_GLOBAL_BASE_ADDR
Description	Specifies the start address of DSPR for core <x> where x=0 up to maximum number of cores available in hardware.
Verification method	The macro is generated based on end address of DSPR for core<x> in the device. If the

	PSPR for the core is not available, the macro is generated as 0. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate McalLib_Cfg.h file with
	• DSPR0 start address is 1879048192
	• DSPR1 start address is 1610612736
	• DSPR2 is not available
	• DSPR3 is not available
	• DSPR4 start address is 805306368
	• DSPR5 start address is 268435456
		`#define MCAL_DSPR0_GLOBAL_BASE_ADDR (1879048192)`
		`#define MCAL_DSPR1_GLOBAL_BASE_ADDR (1610612736)`
		`#define MCAL_DSPR2_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_DSPR3_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_DSPR4_GLOBAL_BASE_ADDR (805306368)`
		`#define MCAL_DSPR5_GLOBAL_BASE_ADDR (268435456)`

	• DSPR0 start address is 1610612736
	• DSPR1 is not available
	• DSPR2 start address is 1342177280
	• DSPR3 start address is 1073741824
	• DSPR4 is not available
	• DSPR5 is not available
		`#define MCAL_DSPR0_GLOBAL_BASE_ADDR (1610612736)`
		`#define MCAL_DSPR1_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_DSPR2_GLOBAL_BASE_ADDR (1342177280)`
		`#define MCAL_DSPR3_GLOBAL_BASE_ADDR (1073741824)`
		`#define MCAL_DSPR4_GLOBAL_BASE_ADDR (0)`
		`#define MCAL_DSPR5_GLOBAL_BASE_ADDR (0)`

1.1.13 Macro: MCAL_DSPR<x>_GLOBAL_END_ADDR

Table 13 MCAL_DSPR<x>_GLOBAL_END_ADDR

Name	MCAL_DSPR<x>_GLOBAL_END_ADDR	
Description	Specifies the end address of DSPR for core 0.	
Verification method	The macro is generated based on end address of DSPR for core <x> where x=0 up to maximum number of cores available in hardware. *Note: The macro is not user configurable.*	
Example(s)	**Action**	**Generated output**

	Generate McalLib_Cfg.h file with
	• DSPR0 end address is 1879293951
		`#define MCAL_DSPR0_GLOBAL_END_ADDR (1879293951)`
		`#define MCAL_DSPR1_GLOBAL_END_ADDR (1610858495)`

	• DSPR1 end address is 1610858495
	• DSPR2 is not available
	• DSPR3 end address is 1073840127
	• DSPR4 end address is 805404671
	• DSPR5 end address is 268533759
		`#define MCAL_DSPR2_GLOBAL_END_ADDR (0)`
		`#define MCAL_DSPR3_GLOBAL_END_ADDR (1073840127)`
		`#define MCAL_DSPR4_GLOBAL_END_ADDR (805404671)`
		`#define MCAL_DSPR5_GLOBAL_END_ADDR (268533759)`

	• DSPR0 end address is 1610858495
	• DSPR1 is not available
	• DSPR2 end address is 1342275583
	• DSPR3 is not available
	• DSPR4 is not available
	• DSPR5 is not available
		`#define MCAL_DSPR0_GLOBAL_END_ADDR (1610858495)`
		`#define MCAL_DSPR1_GLOBAL_END_ADDR (0)`
		`#define MCAL_DSPR2_GLOBAL_END_ADDR (1342275583)`
		`#define MCAL_DSPR3_GLOBAL_END_ADDR (0)`
		`#define MCAL_DSPR4_GLOBAL_END_ADDR (0)`
		`#define MCAL_DSPR5_GLOBAL_END_ADDR (0)`

1.1.14 Macro: MCAL_MASTER_COREID

Table 14 MCAL_MASTER_COREID

Name	MCAL_MASTER_COREID	
Description	Indicates the CPU ID which is considered as the master core.	
Verification method	The macro is generated based on 'ResourceMMcalConfig/ResourceMMasterCore'.	
Example(s)	**Action**	**Generated output**
	Generate McalLib_Cfg.h file with ResourceMMasterCore = 1	`#define MCAL_MASTER_COREID (1)`
	Generate McalLib_Cfg.h file with ResourceMMasterCore = 5	`#define MCAL_MASTER_COREID (5)`

Revision history

Major changes since the last revision

Date	Version	Description
2020-10-27	3.0	MCALLIB driver chapter moved from MC-ISAR_TC3xx_Config_Verification_Manual_BASIC.pdf to this document
2019-07-19	2.0	Updated revision history
2019-02-20	1.10.0_1.0	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-10-27
Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
Doc_Number

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”) .

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.