AUTOSAR MCAL R4.0.3
User's Manual

FLS Driver Component Ver.1.0.2

Generation Tool User's Manual

Target Device:

RH850/P1x-C

Renesas Electronics

www.renesas.com

Rev.1.00 Feb 2017

Abbreviations and Acronyms

Abbreviation / Acronym	Description
AUTOSAR	AUTomotive Open System ARchitecture
BSWMDT	Basic Software Module Description Template
DEM	Diagnostic Event Manager
ECU	Electronic Control Unit
FLS	Flash Driver
Id	Identifier
MCAL	Microcontroller Abstraction Layer
MCU	MicroController Unit
XML	eXtensible Mark-up Language
ARXML	AUTOSAR XML

Definitions

Terminology	Description
BSWMDT File	This file is the template for the Basic Software Module Description.
Configuration XML File	This file contains the setting of command line options.
ECU Configuration Description File	Input file to FLS Driver Generation Tool. It is generated by ECU Configuration Editor.
Sl.No	Serial Number.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

The FLS Software component provides the service for initializing the whole FLS structure of the microcontroller.

The FLS Software Component comprises of two sections as Embedded Software and the MCAL Code Generator Tool to achieve scalability and configurability.

The document describes the FLS module specific inputs and outputs of the MCAL Code Generator Tool that is the common code generator engine used for the generation of the configuration code for all MCAL modules. The document is intended for the developers of ECU software using Application Programming Interfaces provided by the AUTOSAR. MCAL Code Generator Tool is a command line tool that extracts information from ECU Configuration Description File and generates FLS Software C Header files and Source Files (Fls_Cbk.h, Fls_Cfg.h, Fls_Hardware.h, Fls_Hardware.c and Fls_PBcfg.c).

This document contains information on the options, input and output files of the MCAL Code Generator Tool. In addition, this manual covers a step-by-step procedure for the usage of tool.

1.1   Document Overview

This user manual is organized as given in the table below:

Section	Contents
Section 1 (Introduction)	Provides an introduction to the document and explains how information is organized in this manual.
Section 2 (Reference)	Provides a list of documents referred while developing this document.
Section 3 (Code Generation Overview)	Provides the component overview of FLS Driver.
Section 4 (Input Files)	Provides information about ECU Configuration Description File.
Section 5 (Output Files)	Explains the output files that are generated by the MCAL Code Generator Tool.
Section 6 (Precautions)	Contains precautions to be taken during configuration of ECU Configuration Description File.
Section 7 (User Configuration Validation)	Describes about user configuration validation done by the MCAL Code Generator Tool.
Section 8 (Configuration Overview)	Describes all the containers and configurable parameters.
Section 9 (Messages)	Explain the steps to generate the output files.

Chapter 2 Reference

2.1 Reference Documents

The following table lists the documents referred to develop this document:

Sl.No.	Title	Version
1	AUTOSAR_SWS_FlashDriver.pdf	3.2.0
2	MCAL_CodeGenerator_Tool_UserManual.pdf	1.7
3	R20UT3642EJ0100-AUTOSAR.pdf	1.0.2

2.2 Trademark Notice

Microsoft and Windows are trademarks/registered trademarks of Microsoft Corporation.

Chapter 3 Code Generation Overview

Code Generation overview is shown below.
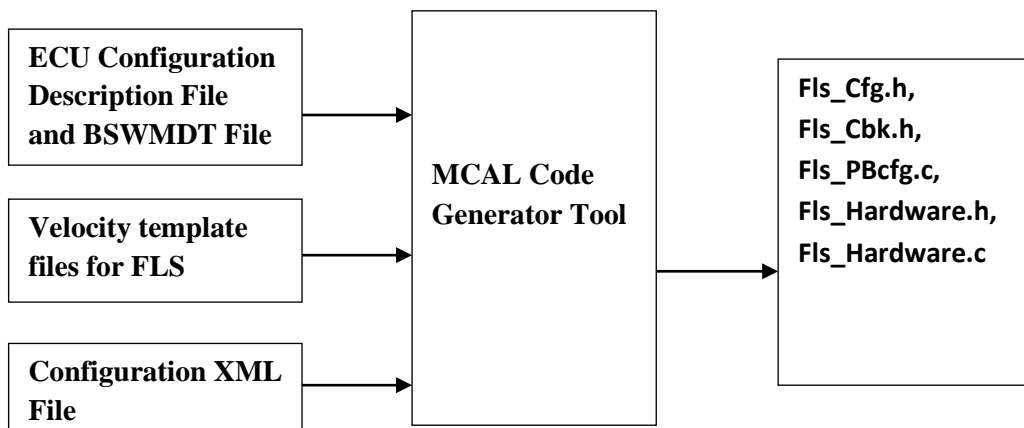
Figure 3-1   Overview of Code Generation

- ECU Configuration Description File (.arxml):
  This file will contain FLS Driver specific configuration information. This file should be generated by AUTOSAR specified Configuration Editor.

- BSWMDT File (.arxml):
  MCAL Code Generator Tool uses "Common Published Information" from FLS module specific BSWMDT File. FLS module specific BSWMDT File should not be updated manually since it is "Static Configuration" file.

- Velocity template files:
  Fls_PBcfg_c.vm, Fls_Cfg_h.vm, Fls_Cbk_h.vm, Fls_Hardware_c.vm, Fls_Validate.vm, Fls_Hardware_h.vm
  They are interpreted by the MCAL Code Generator Tool in order to provide user input validation and generate the final output file needed by the AUTOSAR configuration chain. They are the "logic" of the Code Generator.

- Configuration XML File (.xml):
  This file is used to specify which velocity template to use and their location and the name of the output file generated.

For the error free input file, the tool generates the following output files: Fls_Cfg.h, Fls_PBcfg.c, Fls_Hardware.c, Fls_hardware.h and Fls_Cbk.h and displays appropriate context sensitive error messages for wrong input and exits.

ECU Configuration Description File can be created or edited using ECU Configuration Editor.

13

Concept of execution for MCAL Code Generator Tool
is as follows:
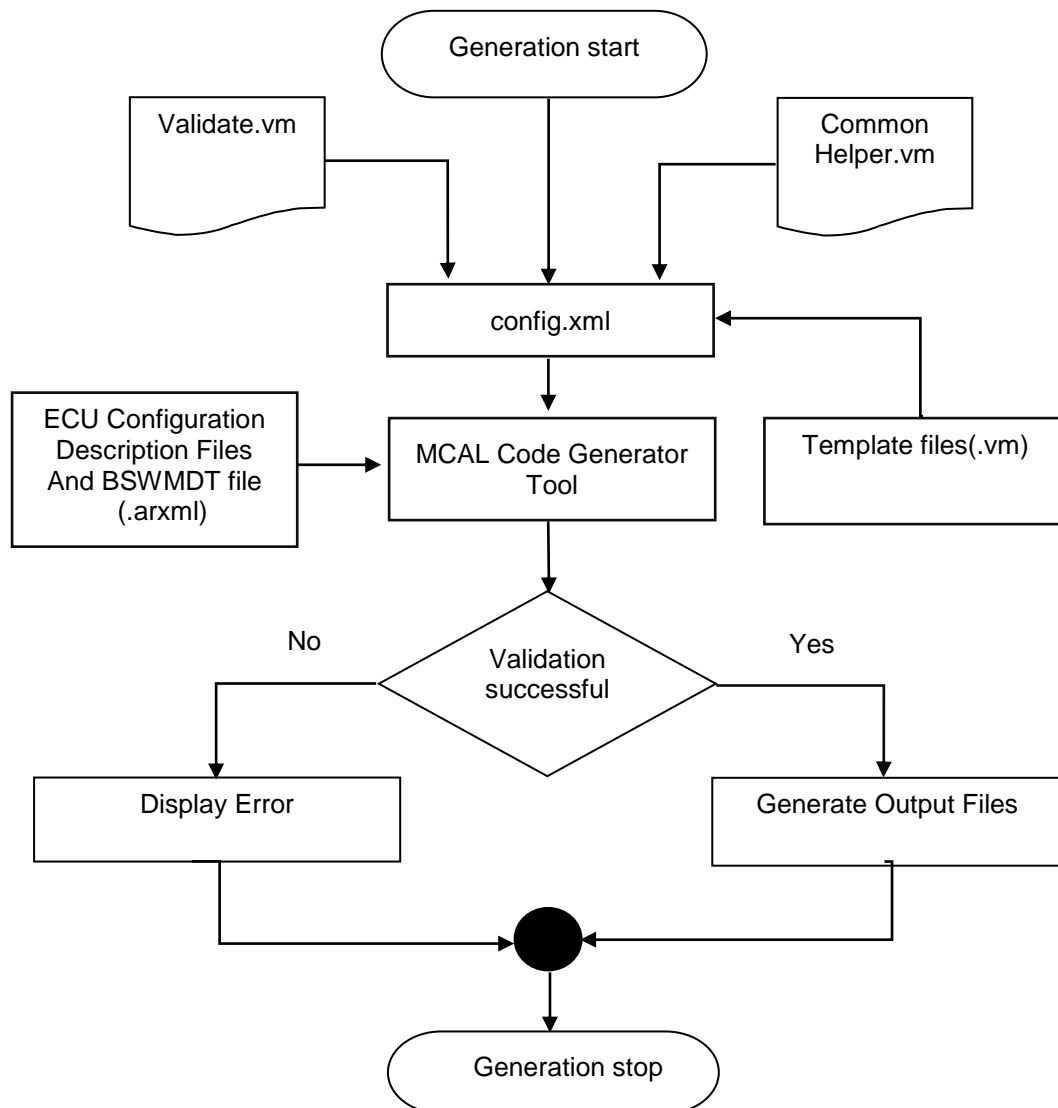
Figure 3-2   Flow-Diagram of Code Generation

The module "Validate" will validate the configuration (contents of ECU
Configuration Description File(s) as input). If there are incorrect values or
incorrect dependencies, the tool will display error, warning and information
messages. In case of errors, the tool will abort the execution.

Fls_Cfg_h.vm will generate compiler switch necessary to the AUTOSAR
Configuration chain and vendor specific parameters.

Fls_PBcfg_c.vm will generate structures necessary to the AUTOSAR
Configuration chain and vendor specific parameters

Fls_Cbk_h.vm will generate the callback notification function prototypes.

 Fls_Hardware_h.vm / Fls_Hardware_c.vm will generate hardware related info
(defines number of actual instances / channels used / structure to access to
the I/O mapped peripheral)

Chapter 4 Input Files

MCAL Code Generator Tool will accept the config.xml file which has paths to the code generator template files for generating FLS Driver files. MCAL Code Generator Tool need ECU Configuration Description File(s) and BSWMDT File as inputs to generate FLS Driver specific source files. Hence ECU Configuration Description File should contain configuration of FLS Driver module. MCAL Code Generator Tool ignores any other AUTOSAR component configured in the ECU Configuration Description File. ECU Configuration Description File can be generated using configuration editor.

ECU Configuration Description File must comply with AUTOSAR standard ECU Configuration Description File format.

Remark   The detailed explanation about the parameters and containers are found in Parameter Definition File.

Chapter 5 Output Files

MCAL Code Generator Tool generates configuration details in C Header and C Source files (Fls_Cbk.h, Fls_Cfg.h, Fls_Hardware.c, Fls_Hardware.h, Fls_PBcfg.c).

The content of each output file is given in the table below:

Table 5-1   Output Files Description

Output File	Details
Fls_Cbk.h	This file contains call-back functions prototype declarations.
Fls_Cfg.h	This file contains pre-compile time parameters.
Fls_PBcfg.c	This file contains post-build time parameters.
Fls_Hardware.c	This file contains hardware information.
Fls_Hardware.h	This file contains hardware information.

Remark   Output files generated by MCAL Code Generator Tool should not be modified or edited manually.

Chapter 6 Precautions

- ECU Configuration Description File and BSWMDT File must comply with AUTOSAR standard for R4.0.3 ECU Configuration Description File and BSWMDT File respectively.

- ECU Configuration is dependent on DEM, MCU module description file.

- Configuration XML File should contain the file extension '.xml'.

- config.xml file should be configured with the template file location and output file location.

- The input file must contain FLS Driver Dependent configurations.

- All the function names and the string values configured should follow C syntax for variables. It can only contain alphanumeric characters and "_". It should start with an alphabet.

- If the output files generated by MCAL Code Generator Tool are modified externally, then they may not produce the expected results or may lead to error/warning/Information messages.

- Short Name for a container should be unique within a name space.

- An error free ECU Configuration Description File generated from configuration editor has to be provided as input to the MCAL Code Generator Tool. Otherwise Tool may not produce the expected results or may lead to errors/warnings/information messages.

Remark: Please refer the AUTOSAR_FLS_Component_UserManual.pdf (R20UT3641EJ0100-AUTOSAR.pdf) for deviations from AUTOSAR Specifications, if any.

Chapter 7 User Configuration Validation

This section provides help to analyze the error, warning and information messages displayed during the execution of MCAL Code Generator Tool. It ensures conformance of input file with syntax and semantics. It also performs validation on the input file for correctness of the data.

For more details on list of Error/Warning/Information messages that are displayed as a result of input file(s) validation, refer Chapter 9 "*Messages*".

The MCAL Code Generator Tool displays error or warning or information when the user has configured incorrect inputs. The format of Error/Warning/Information message is as shown below.

<message_type>_<vendor_id>_<module_id>_<message_id>:<message_cont ent>.

where,

- <message_type>   : ERR/ WRNINING /INFO   (Error/Warning/Information)

- < vendor_id >   : vendor Id = 59

- < module_id >   : 092 - FLS Driver Module id (92) for user
                           configuration checks.

- < Message_id.>   : 001-026

- <message_content>:   Message content provides information
                        about error or warning or information displayed
                        when the user has configured incorrect inputs.

                            File Name' and 'Path' need not be present for all
                            Error/Warning/Information messages.

                            File Name: Name of the file in which the error has
                            occurred.

                            Path: Absolute path of the container in which the
                            Parameter that caused the message is present.

Chapter 8 Configuration Overview

8.1 Container Overview
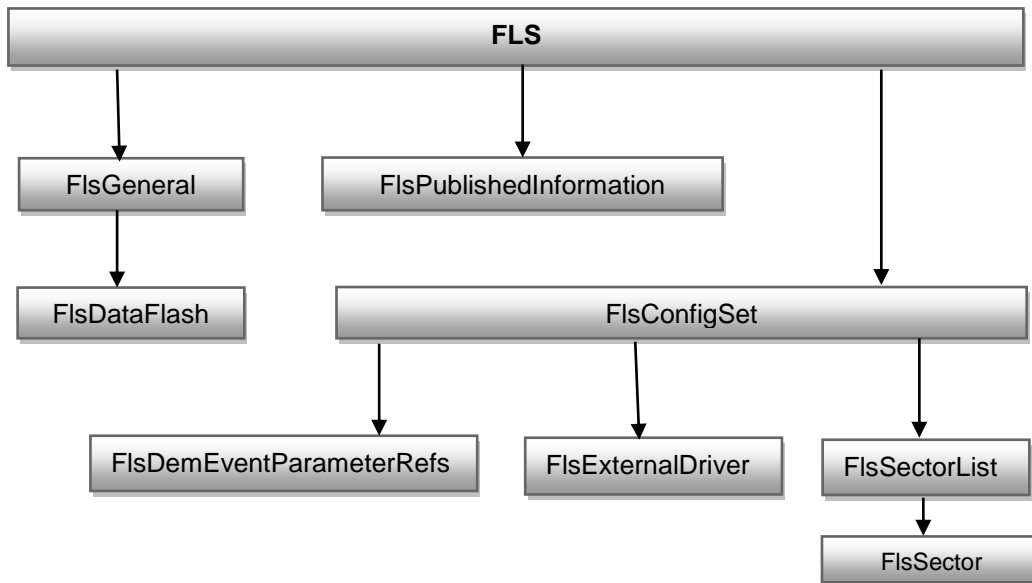
Figure 8-1 Container Overview

8.2 Configurable Parameters

Table 8-1   Configurable Parameters

Container Name	Parameter Name	Parameter Type	Parameter Range	Parameter Description
FlsGeneral	FlsCancelApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_Cancel function.
	FlsAcLoadOnJobStart	Boolean	TRUE / FALSE	This parameter specifies the flash driver shall load the flash access code to RAM whenever an erase or write job is started and unload (overwrite) it after that job has been finished or canceled. This parameter is not used for the implementation.
	FlsUseInterrupts	Boolean	TRUE / FALSE	This parameter specifies job processing triggered by hardware interrupt. This parameter is not used for the implementation.

	FlsCompareApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_Compare function.
	FlsDevErrorDetect	Boolean	TRUE / FALSE	This parameter specifies pre-processor switch to enable and disable development error detection.
	FlsGetJobResultApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_GetJobResult function.
	FlsGetStatusApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_GetStatus function.
	FlsSetModeApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_SetMode function.
	FlsVersionInfoApi	Boolean	TRUE / FALSE	This parameter specifies pre-processor switch to enable / disable the API to read out the modules version information.
	FlsVersionCheckExternalModules	Boolean	TRUE / FALSE	Enable / disable AUTOSAR Version check for inter-module dependencies.
	FlsBlankCheckApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_BlankCheck function.
	FlsReadImmediateApi	Boolean	TRUE / FALSE	This parameter compiles switch to enable and disable the Fls_ReadImmediate function.
	FlsCriticalSectionProtection	Boolean	TRUE / FALSE	This parameter specifies if the FLS driver CPU load can be reduced by disabling enter/exit critical section functionality by adding a precompiled configuration parameter to the FLS driver configuration.
	FlsFaciEccCheck	Boolean	TRUE / FALSE	This parameter is used to enable or disable the check to ensure the ECC error interrupt enable FACI register settings during FLS module initialization.If the user configures this parameter as true, then user should configure "FLS_E_ECC_FAILED" parameter in "FlsConfigSet" container.

	FlsWriteVerify	Enumeration	FLS_DISABLED/ FLS_INIT_ONLY/ FLS_INIT_AND_RU NTIME	This parameter enables selection and support of INIT_ONLY or INIT_AND_RUNTIME register read-back functionality of FACI and ECC registers.
	FlsWriteVerifyErrorInterface	Boolean	NULL	This parameter is used to configure the name of the error notification which shall be called with the ApiID and ErrorId when there is a Write Verify error.
	FlsLoopCount	Integer	32-255	This parameter is used to avoid the risk of endless loops in FLS driver. The loop count can be minimum 32 to maximum 255
	FlsUseWVErrorInterface	Boolean	TRUE / FALSE	This parameter specifies whether user can configure error notification interface instead of DEM error for Write-Verify functionality.
	FlsDriverIndex	Integer	0-254	This parameter specifies index of the driver, used by FEE. This parameter is not used for implementation.
	FlsDeviceName	Enumeration	RF701370A, RF701372, RF701373, RF701374	This parameter contains the supported device name to identify the device specific C header file
	FlsVirtualBoundaryAddress	integer	65536 for R7F701373 131072 for R7F701372 196608 for R7F701370A	This parameter specifies the virtual boundary address for Data Flash and CodeFlash. This parameter is used only for the validation purpose of virtual address range of Data Flash and Code Flash.
	FlsBaseAddress	integer	0	This parameter specifies the flash memory start address. This parameter is not used for implementation.
	FlsTotalSize	integer	0	This parameter specifies the total amount of flash memory in bytes. This parameter is not used for implementation.
	FlsTimeoutMonitoring	Boolean	TRUE / FALSE	This Parameter compiles switch to enable or disable the FlsTimeout Supervision.
	FlsWriteTime	float	0.000530	
FlsPublishedInform	FlsErasedValue	Integer	4294967295	

ation	FlsBlankCheckTime	float	0.000213	
	FlsAcLocationErase	Integer	0	Position in RAM, to which the erase flash access code has to be loaded. This parameter is not used for implementation. Only relevant if the erase flash access code is not position independent. If this information is not provided it is assumed that the erase flash access code is position independent and that therefore the RAM position can be freely configured.
	FlsAcLocationWrite	Integer	0	Position in RAM, to which the write flash access code has to be loaded. This parameter is not used for implementation. Only relevant if the write flash access code is not position independent. If this information is not provided it is assumed that the write flash access code is position independent and that therefore the RAM position can be freely configured.
	FlsAcSizeErase	Integer	0	Number of bytes in RAM needed for the erase flash access code. This parameter is not used for implementation.
	FlsAcSizeWrite	Integer	0	Number of bytes in RAM needed for the write flash access code. This parameter is not used for implementation.
	FlsSuspendTime	Float	0.000300	Maximum time (in seconds) to perform suspend operation.
	FlsCancelTime	Float	0.000050	Maximum time (in seconds) to do cancel of ongoing job for data flash.
	FlsExpectedHwId	string	NULL	Unique identifier of the hardware device that is expected by this driver (the device for which this driver has been implemented). This parameter is not used for the implementation. Only relevant for external flash drivers.

	FlsEraseTime	Float	0.004	Maximum time (in milliseconds) to erase one complete data flash block.
	FlsSpecifiedEraseCycles	Integer	125000	Number of erase cycles specified for the flash device. This parameter is applicable only for Data flash. This parameter is not used for implementation.
FlsConfigSet	FlsMaxReadNormalMode	Integer	4-8192	This parameter specifies the maximum number of bytes to read or compare in one cycle of the flash driver's job processing function in normal mode. This parameter is applicable for data flash.
	FlsMaxWriteNormalMode	Integer	4-8192	This parameter specifies the maximum number of bytes to write in one cycle of the flash driver's job processing function in normal mode. This parameter is applicable for data flash.
	FlsMaxEraseNormalMode	Integer	64-8192	This parameter specifies the maximum number of bytes to erase in one cycle of the flash driver's job processing function in normal mode. This value should be multiple of 64 bytes. This parameter is applicable only for Data Flash.
	FlsAcErase	Integer	0	This parameter specifies address offset in RAM to which the erase flash access code shall be loaded. Used as function pointer to access the erase flash access code. This parameter is not used for implementation.
	FlsAcWrite	Integer	0	This parameter specifies address offset in RAM to which the write flash access code shall be loaded. Used as function pointer to access the write flash access code. This parameter is not used for implementation.
	FlsDefaultMode	Enumeration	NOT_USED	This parameter is the default FLS device mode after initialization. This parameter is not used for implementation.

	FlsJobEndNotification	string	-	This parameter mapped to the job end notification routine provided by some upper layer module, typically the Fee module.
	FlsJobErrorNotification	string	-	This parameter mapped to the job error notification routine provided by some upper layer module, typically the Fee module.
	FlsMaxReadFastMode	Integer	0	This parameter specifies the maximum number of bytes to read or compare in one cycle of the flash driver's job processing function in fast mode. This parameter is not used for implementation.
	FlsMaxWriteFastMode	Integer	0	This parameter specifies the maximum number of bytes to write in one cycle of the flash driver's job processing function in fast mode. This parameter is not used for implementation.
	FlsCallCycle	Float	0-1	This parameter specifies the cycle time of calls of the flash driver's main function (in seconds). The value is calculated for 160MHz frequency.
	FlsTimeOutCountValue	Integer	1-1000000	This parameter specifies the time out count required for erase, write and blank check operations during interrupt mode.Incorrect configuration of this parameter shall lead to erroneous operations of FLS Driver
	FlsProtection	Integer	0	This parameter uses the erase/write protection settings. (Only relevant if supported by hardware). This parameter is not used for implementation.
FlsSector	FlsSectorIndex	Integer	0-2	This parameter specifies unique index value for data flash sector.
	FlsNumberOfSectors	Integer	1-3072	This parameter specifies number of continuous sectors with identical values for FlsSectorSize and FlsPageSize.
	FlsPageSize	integer	4	This parameter specifies size of one page of this sector.

	FlsSectorSize	integer	64	This parameter specifies size of this sector in bytes.
	FlsSectorStartaddress	integer	0 - 196607	This parameter specifies the start address of this sector. This parameter specify the virtual start address. Implementation Type: Fls_AddressType.
FlsDataFlash	FlsDFBaseAddress	integer	4280287232	This parameter specifies the data flash memory start address. This parameter is applicable only for Data flash.
	FlsDFBlockSize	integer	64	This parameter specifies the block size of data flash memory in bytes. This parameter is applicable only for Data flash.
	FlsDFTotalBlocks	integer	1024 for R7F701373 and RF701374 2048 for R7F701372 3072 for R7F701370A	This parameter specifies the total number of data flash blocks. There are 512 64-Byte physical blocks in Data Flash. This parameter is applicable only for Data flash.
	FlsEccSedNotification	function	-	This parameter mapped to Single-bit error (SED) notification routine provided by some upper layer module.
	FlsDataFlashSize	integer	-	65536-196608
	FlsResumeApi	Boolean	TRUE/FALSE	This parameter compiles switch to enable and disable the Fls_Resume function.
	FlsSuspendApi	Boolean	TRUE/FALSE	This parameter compiles switch to enable and disable the Fls_Suspend function.
	FlsEccDedNotification	function	-	This parameter mapped to Double-bit error (DED) notification routine provided by some upper layer module.
	FlsFdlCpuFrequency	Reference	Reference to McuPLLClkSetting container of MCU module.	Reference to the CPU subsystem frequency configuration, which is set in the MCU driver configuration. This reference from MCU informs the programming HW frequency to the programming Firmware. The clock value in MCU must be according to the clock supply setting of the CPU Subsystem.

FlsDemEventParameterRefs	FLS_E_ERASE_FAILED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash erase failed (HW)" has occurred.
	FLS_E_READ_FAILED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash read failed (HW)" has occurred.
	FLS_E_WRITE_FAILED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash write failed (HW)" has occurred.
	FLS_E_UNEXPECTED_FLASH_ID	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Expected hardware ID not matched"
	FLS_E_READ_FAILED_DED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash read failed (HW) due to double bit ECC error" has occurred.
	FLS_E_INT_INCONSISTENT	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when Interrupt consistency error was detected.
	FLS_E_REG_WRITE_VERIFY	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash register write verification failed (HW)" has occurred.
	FLS_E_HW_FAILURE	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash Hardware Failure (HW)" has occurred.
	FLS_E_ECC_FAILED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Incorrect FACI ECC register settings during initialisation (HW)" has occurred.
	FLS_E_COMPARE_FAILED	Reference	Reference to DemEventId of Dem module.	Reference to the DemEventParameter which shall be issued when the error "Flash compare failed (HW)" has occurred.

Chapter 9 Messages

The messages help to identify the syntax or semantic errors in the ECU Configuration Description File. Hence it ensures validity and correctness of the information available in the ECU Configuration Description File.

The following section gives the list of error, warning and information messages displayed by the MCAL Code Generator Tool.

9.1 Error Messages

ERR_59_92_001: Parsing of Fls or Dem module is incorrect.

This error occurs when parsing of Fls or Dem module is incorrect.

ERR_59_92_003: References path of Parameter <Error ID> is not correct in the container FlsDemEventParameterRefs

This error occurs when the References path for the Parameters <Error ID> is not correct in the container FlsDemEventParameterRefs.

Table 9-1   Error ID related to error ERR_59_92_003

Sl.No.	Error ID
1.	FLS_E_COMPARE_FAILED
2.	FLS_E_ERASE_FAILED
3.	FLS_E_READ_FAILED
4.	FLS_E_READ_FAILED_DED
5.	FLS_E_WRITE_FAILED
6.	FLS_E_HW_FAILURE
7.	FLS_E_ECC_FAILED
8.	FLS_E_REG_WRITE_VERIFY

ERR_59_92_004: The value configured for the parameter FlsJobEndNotification/ FlsJobErrorNotification should follow C syntax <[a-zA-Z][a-zA-Z0-9_]>

This error occurs when the value configured for the parameters FlsJobEndNotification/ FlsJobErrorNotification in the path /Renesas/EcucDefs_Fls/Fls/FlsConfigSet are other than syntax <[a-zA-Z][a-zA-Z0-9_]>

ERR_59_92_006: The parameters FlsJobEndNotification and FlsJobErrorNotification in the container FlsConfigSet should be configured with different values

This error occurs when the parameters FlsJobEndNotification and FlsJobErrorNotification in the path /Renesas/EcucDefs_Fls/Fls/FlsConfigSet are configured with same values.

ERR_59_92_016: References path configured for the parameter FlsFdlCpuFrequency is not correct in the path /Renesas/EcucDefs_Fls/Fls/FlsGeneral/FlsDataFlash.

This error occurs when references path configured for the parameter FlsFdlCpuFrequency is not correct in the path /Renesas/EcucDefs_Fls/Fls/FlsGeneral/FlsDataFlash.

ERR_59_92_017: The value configured for the parameter FlsSectorStartaddress in the container FlsSector should be within the range of <range of value>.

This error occurs, if the value configured for the parameter FlsSectorStartaddress is not within the below range.

Table 9-2   Parameters and Container related to error ERR_59_92_017

Range of value for FlsSectorStartaddress
0 to (value configured for (FlsVirtualBoundaryAddress – DFTotalSize))

ERR_59_92_018: The value configured for the parameter FlsVirtualBoundaryAddress in the container FlsGeneral should be <Value>, since the Device Name is <Device Name>.

This error occurs when the value configured for the parameter FlsVirtualBoundaryAddress in the container FlsGeneral is other than <Value> when the Device Name is <Device Name>.

Table 9-3   Device Name related to error error ERR_59_92_018

Sl.No.	Device Name	Value In bytes
1.	R7F701370A	195904
2.	R7F701371	195904
3.	R7F701372	130368
4.	R7F701373	64832
5.	R7F701374	64832

ERR_59_92_019: The value configured for the parameter FlsSectorStartaddress in the container FlsSector should be multiple of FlsSectorSize[64].

This error occurs, if the value configured for the parameter FlsSectorStartaddress is not a multiple of FlsSectorSize[64].

ERR_59_92_020: The value configured for the parameter FlsDFTotalBlocks in the container FlsDataFlash should be <Value>, since the Device Name is <Device Name>.

This error occurs when the value configured for the parameter FlsDFTotalBlocks in the container FlsDataFlash is other than <Value>, when the Device Name is <Device Name>

Table 9-4   Device Name related to error ERR_59_92_020

Sl.No.	Device Name	Value In **bytes**
1.	R7F701370A	3061
2.	R7F701371	3061
3.	R7F701372	2037
4.	R7F701373	1013
5.	R7F701374	1013

ERR_59_92_023: The value configured for the parameter FlsUseInterrupts in the container FlsGeneral should be FALSE, since interrupt based job handling is not supported by P1x-C FLS driver.

This error occurs when the value configured for the parameter FlsUseInterrupts in the container FlsGeneral isn't FALSE when interrupt based job handling is not supported by P1x-C FLS driver.

ERR_59_92_026: The device variant selected: <Device Name> is not supported by the P1X-C family.

This error occurs when the device variant selected: <Device Name> is not supported by the P1X-C family.

Table 9-5   Device Name related to error ERR_59_92_026

Sl.No.	Device Name
1.	R7F701370A
2.	R7F701371
3.	R7F701372
4.	R7F701373
5.	R7F701374

ERR_59_92_027: The value configured for the parameter FlsCallCycle in the container FlsConfigSet<config_index> **should be same across the multiple configuration set.**

This error occurs when value configured for the parameter FlsCallCycle in the container FlsConfigSet are different across the multiple configuration set.

ERR_59_92_028: The value configured for the parameter FlsTimeOutCountValue in the container FlsConfigSet<config_index> **should be same across the multiple configuration set.**

This error occurs when value configured for the parameter FlsTimeOutCountValue in the container FlsConfigSet are different across the multiple configuration set.

ERR_59_92_029: The value configured for the parameter FlsWriteVerifyErrorInterface in the container FlsConfigSet<config_index> **shall not be Empty or NULL,since FlsUseWVErrorInterface is configured as True.**

This error occurs when value configured for the parameter FlsWriteVerifyErrorInterface in the container FlsConfigSet shall not be Empty or NULL,since FlsUseWVErrorInterface is configured as True.

ERR_59_92_1008: The Value of the Boolean Parameter <Parameter Name> configured is not a valid one.

This error occurs when any of the Boolean Parameters mentioned below are not configured in ECU Configuration Description File. The list of boolean parameters with respect to container is listed below:

Table 9-6   Parameters and Container related to error ERR_59_92_1008

Parameter Name	Container Name
FlsDevErrorDetect	**FlsGeneral**
FlsCompareApi	
FlsCancelApi	
FlsSetModeApi	
FlsGetStatusApi	
FlsGetJobResultApi	
FlsBlankCheckApi	
FlsReadImmediateApi	
FlsVersionInfoApi	
FlsVersionCheckExternalModules	
FlsCriticalSectionProtection	
FlsFaciEccCheck	
FlsTimeoutMonitoring	
FlsUseWVErrorInterface	
FlsSuspendApi	**FlsDataFlash**
FlsResumeApi	

ERR_59_92_031:   The   value   configured   for   the   parameter FlsNumberOfSectors in the container FlsSector should be within 1-max range for the device <Device Name>.

This error occurs when value configured for the parameter FlsNumberOfSectors in the container FlsSector is not within 1-max range for the device <Device Name>.

Table 9-7   Device Name related to error ERR_59_92_031

Sl.No.	Device Name	Value In bytes
1.	R7F701370A	3061
2.	R7F701371	3061
3.	R7F701372	2037
4.	R7F701373	1013
5.	R7F701374	1013

ERR_59_92_032: If the parameter FlsWriteVerify is configured as FLS_INIT_ONLY   or   FLS_INIT_AND_RUNTIME,   then   the   parameter 'FLS_E_REG_WRITE_VERIFY' in the container FlsDemEventParameterRefs should be configured.

This error will occur if no reference path is provided for the DEM parameter FLS_E_REG_WRITE_VERIFY in the container FlsDemEventParameterRefs when

the parameter FlsUseWVErrorInterface is configured as true.

ERR_59_92_033: The notification function configured for the parameter FlsEccDedNotification and FlsEccSedNotification in the FlsGeneral container should be unique.

This error occurs when the notification function configured for the parameter FlsEccDedNotification and FlsEccSedNotification in the FlsGeneral container should be unique.

ERR_59_92_034: The values configured for the parameters EccSedNotification and EccDedNotification in the container FlsDataFlash and the value configured for the parameter <Parameter Name> in any of the ConfigSet container should be unique, which is not followed in FlsConfigSet<config_index>

This error occurs when the values configured for the parameters EccSedNotification and EccDedNotification in the container FlsDataFlash and the value configured for the parameter <Parameter Name> in any of the ConfigSet container is not unique.

Table 9-8   Parameters and Container related to error ERR_59_92_034

Parameter Name	Container Name
FlsJobEndNotification	**FlsGeneral**
FlsJobErrorNotification	

ERR_59_92_035: Mandatory Parameter <Parameter Name> in the container FlsSector of FlsConfigSet<config_index> should be configured.

This error occurs when mandatory Parameter <Parameter Name> in the container FlsSector of FlsConfigSet<config_index> is not configured.

Table 9-9   Parameters and Container related to error ERR_59_92_035

Parameter Name	Container Name
FlsNumberOfSectors	**FlsSector**
FlsSectorStartaddress	

ERR_59_92_036: The value configured for the parameter <Parameter Name> in the container FlsSector of FlsConfigSet<config_index> should be same for the FlsSectorIndex across the multiple configuration set.

This error occurs when the value configured for the parameter <Parameter Name> in the container FlsSector of FlsConfigSet<config_index> is not same for the FlsSectorIndex across the multiple configuration set.

Table 9-10 Parameters and Container related to error ERR_59_92_036

Parameter Name	Container Name
FlsNumberOfSectors	**FlsSector**
FlsSectorStartaddress	

ERR_59_92_037: User shall not configure multiple sectors. Since data flash is a

monolithic on-chip NV memory with homogeneous block size, it is not required to have multiple sectors with the same sector sizes. This is not followed in FlsConfigSet<config_index>

This error occurs when user configure multiple sectors. Since data flash is a monolithic on-chip NV memory with homogeneous block size, it is not required to have multiple sectors with the same sector sizes.

9.2 Warning Messages

None

9.3 Information Messages

INF_59_092_001: The parameter FlsJobEndNotification in the container FlsConfigSet is not configured

This information occurs when the parameter FlsJobEndNotification in the container FlsConfigSet is not configured.

Sl.No.	Description	Version	Date
1.	Initial Version	1.0.0	12-Aug-2015
2.	Following changes are done: 　1.　Section 8.1 is updated. 　2.　Details related to code flash is removed. 　3.　Chapter 10 updated. 　4.　R Number is added.	1.0.1	13-May-2016
3.	Following changes are done: 　1.　INF_59_092_001 is added in section 9.3 　2.　Chapter 1, Updated Introduction. 　3.　Section 2.1, Updated reference document details. 　4.　Chapter 3, Updated the Heading Code Generation Overview with MCAL Code Generator overview 　5.　Chapter 5, Updated description of output files. 　6.　Chapter 6, Added one more point in precautions. 　7.　Section 8.1, Updated Pre-Compile Configurable Parameters 　8.　Section 8.2, Updated Post Build Time Configurable Parameters 　9.　Chapter 4, Updated description of Input files. 　10.　Removed chapter 9, MCAL Code Generator Tool 　11.　Removed Chapter 11 - Notes, 　12.　Replaced the words Tool, Generator Tool, FLS Driver Generator tool with MCAL Code Generator Tool in this document. 　13.　Chapter 3, Added remark for common MCAL Code Generator Tool user manual. 　14.　Updated Chapter 2, Chapter 3, Chapter 4 and Chapter 8 for maintaining consistency. 　15.　MCAL Code Generator Tool Overview to Code Generation Overview 　16.　Removed parameters from Figure 8-1, Configuration Overview 　17.　Removed Error Messages from chapter 9. 　18.　Added Error Messages in Chapter 9. 　19.　Removed parameters FlsInterruptConsistencyCheck and FlsFaciUnit from section 8.2	1.0.2	27-Feb-2017

RENESAS

Renesas Electronics Corporation

http://www.renesas.com

AUTOSAR MCAL R4.0.3

User's Manual