Integration Manual

for MPC5634M MCU Driver

Document Number: IM14MCUASR3.0R2.0.0
Rev. 1.1

freescale™

Contents

Section Number **Title** **Page**

Chapter 1
Revision History

Table 1-1. Revision History

Revision	Date	Author	Description
1.0	07 - Feb - 2011	Rosario Anchini	First released version
1.1	12 - Dec - 2011	Francesco Mazzarella	Configuration parameters and memory allocation updated

Chapter 2
Introduction

This integration manual describes the integration requirements for MCU Driver for MPC5634M microcontrollers.

2.1 Supported Derivatives

The software described in this document is intented to be used with the following microcontroller devices of Freescale Semiconductor .

Table 2-1. MPC5634M Derivatives

Freescale Semiconductor	mpc5634m_bga208, mpc5634m_qfp144, mpc5634m_qfp176

All of the above microcontroller devices are collectively named as MPC5634M .

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 type: Bold is used for important terms, notes and warnings.

Italic font: Italic typeface is 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.

2.4 Acronyms and Definitions

Table 2-2. Acronyms and Definitions

Term	Definition
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
ASM	Assembler
BSMI	Basic Software Make file Interface
CAN	Controller Area Network
DEM	Diagnostic Event Manager
DET	Development Error Tracer
C/CPP	C and C++ Source Code
VLE	Variable Length Encoding
N/A	Not Applicable
MCU	Micro Controller Unit

2.5 Reference List

Table 2-3. Reference List

#	Title	Version
1	AUTOSAR 3.0MCU Driver Software Specification Document.	V2.2.0 R3.0 Rev 0001
2	MPC5634M Reference Manual	Rev. 6, 4 October 2011

Chapter 3
Building the Driver

This section describes the source files and various compilers, linker options used for building the Autosar MCU driver for Freescale SemiconductorMPC5634M . It also explains the EB Tresos Studio plugin setup procedure.

3.1 Build Options

The MCU driver files are compiled using

- GHS 5.2.4
- DIAB 5_8_0_02 wind00198363 20100511 123238
- CW Version 4.3 build 182

The compiler, linker flags used for building the driver are explained below:

Note

The TS_T2D14M20I0R0 plugin name is composed as follow:

TS_T = Target_Id

D = Derivative_Id

M = SW_Version_Major

I = SW_Version_Minor

R = Revision

(i.e. Target_Id = 2 identifies PowerPC architecture and
Derivative_Id = 14 identifies the MPC5634M)

3.1.1 CW Compiler/Linker/Assembler Options

Table 3-1. Compiler Options

Option	Description
-proc Zen	Generates and links object code for Zen processor. The compiler uses unsigned as the default parameter for the -char switch
-lang c	Expects source code to conform to the language specified by the ISO/IEC 9899-1990 ("C90") standard
-opt all	This option is selected all optimization (the same as -opt speed,level=4,intrinsics,noframe)
-common off	Disables moving uninitialized data into a common section
-sdatathreshold 0	Specifies the threshold size (in bytes) for an item considered by the linker to be small data. (The linker stores small data items in the Small Data address space. The compiler can generate faster code to access this data.)
-sdata2threshold 0	Specifies the threshold size (in bytes) for an item considered by the linker to be small constant data. (The linker stores small constant data items in the Small Constant Data address space.)
-vle	Tells the compiler and linker to generate and lay out Variable Length Encoded (VLE) instructions, available on Zen variants of Power Architecture processors
-use_lmw_stmw on	Enables the use of multiple load and store instructions for function prologues and epilogues
-ir	Include the debug information
-ppc_asm_to_vle	Converts regular Power Architecture assembler mnemonics to equivalent VLE (Variable Length Encoded) assembler mnemonics in the inline assembler
-cpp_exceptions off	When on, generates executable code for C++ exceptions. When off, generates smaller, faster executable code
-func_align 4	Specifies alignment of functions in executable code
-sym dwarf-2,full	Generate DWARF-2-conforming debugging information (Debug With Arbitrary Record Format)
-gdwarf-2	Generate DWARF-2-conforming debugging information (Debug With Arbitrary Record Format). The linker ignores debugging information that is not in the Dwarf 1, Dwarf 2 format
-w on	Turns on most warning messages
-r	Compiler should expect function prototypes
-w undefmacro	Issues warning messages on the use of undefined macros in #if and #elif conditionals
-char unsigned	Controls the default sign of the char data type: char data items are unsigned
-nosyspath	Performs a search of both the user and system paths, treating #include statements of the form #include xyz the same as the form #include "xyz"
-fp none	No floating point code generation
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.
-DMCAL_CER_VALIDATION	-D defines a preprocessor symbol for CER Report

Table continues on the next page...

Table 3-1. Compiler Options (continued)

Option	Description
-DMCAL_VERSION_CHECK	-D defines enable the cross check between the AutoSar component Version Numbers
-DMWERKS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the CWpreprocessor symbol.

Table 3-2. Assembler Options

Option	Description
-proc Zen	Generates and links object code for Zen processor. The compiler uses unsigned as the default parameter for the -char switch
-vle	Tells the compiler and linker to generate and lay out Variable Length Encoded (VLE) instructions, available on Zen variants of Power Architecture processors
-sym dwarf-2,full	Generate DWARF-2-conforming debugging information (Debug With Arbitrary Record Format)
-gdwarf-2	Generate DWARF-2-conforming debugging information (Debug With Arbitrary Record Format). The linker ignores debugging information that is not in the Dwarf 1, Dwarf 2 format.

Table 3-3. Linker Options

Option	Description
-proc Zen	Generates and links object code for Zen processor. The compiler uses unsigned as the default parameter for the -char switch
-code_merging all	Removes duplicated functions to reduce object code size
-far_near_addressing	Simplifies address computations to reduce object code size and improve performance
-vle_enhance_merging	Removes duplicated functions that are called by functions that use VLE instructions to reduce object code size
-listdwarf	DWARF debugging information in the linker's map file
-sym dwarf-2,full	Generate DWARF-2-conforming debugging information (Debug With Arbitrary Record Format)
-char unsigned	Controls the default sign of the char data type: char data items are unsigned.

3.1.2 DIAB Compiler/Linker/Assembler Options

Table 3-4. Compiler Options

Option	Description
-tPPCE200Z3VEG:simple	Sets target processor to PPCE200Z3, generates ELF using EABI conventions, All Single Hardware Floating Point (Single precision uses hardware, double precision is mapped to single precision), selects simple environment settings for Startup Module and Libraries
-Xdialect-ansi	Follow the ANSI C standard with some additions
-XO	Enables extra optimizations to produce highly optimized code
-Xsize-opt	Optimize for size rather than speed when there is a choice

Table continues on the next page...

Table 3-4. Compiler Options (continued)

Option	Description
-Xsmall-data=0	Set Size Limit for “small data” Variables to zero.
-Xsmall-const=0	Set Size Limit for “small const” Variables to zero.
-Xno-common	Disable use of the “COMMON” feature so that the compiler or assembler will allocate each uninitialized public variable in the .bss section for the module defining it, and the linker will require exactly one definition of each public variable
-Xnested-interrupts	Allow nested interrupts
-Xalign-functions=4	Align each function on an address boundary divisible by 4
-g	Generate symbolic debugger information. Do most target-independent optimizations. Also, disable most target-dependent optimizations: option -g2 also disables basic reordering and all peephole optimizations.
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xdebug-local-all	Force generation of type information for all local variables
-Xdebug-local-cie	Create common information entry per module
-Xdebug-struct-all	Force generation of type information for all typedefs, struct, union and class types
-Xforce-declarations	Generates warnings if a function is used without a previous declaration
-ee1481	Generate an error when the function was used before it has been declared
-Xforce-prototypes	Generate warnings if a function is used without a previous prototype declaration
-Xmacro-undefined-warn	Generates a warning when an undefined macro name occurs in a #if preprocessor directive
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files
-Xlint	Generate warnings when suspicious and non-portable C code is encountered. Enables all warnings
-ei1604	Suppress the warning messages 1604.
-W:as:,-l	Pass the option “-l” (lower case letter L) to the assembler to get an assembler listing file
-Wa,-Xisa-vle	Instruct the assembler to expect and assemble VLE (Variable Length Encoding) instructions rather than BookE instructions.
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DDIAB	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the DIAB preprocessor symbol.
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.
-DMCAL_CER_VALIDATION	-D defines a preprocessor symbol for CER Report

Table 3-5. Assembler Options

Option	Description
-tPPCE200Z3VEN:simple	Selects target processor: PPCE200Z3, generates ELF using EABI conventions, NO floating point support, selects simple environment settings for Startup Module and Libraries.
-g	Dump the symbols in the global symbol table in each archive file.
-Xisa-vle	Expect and assemble VLE (Variable Length Encoding) instructions rather than Book E instructions. The default code section is named .text_vle instead of .text, and the default code section fill "character" is set to 0x44444444 instead of 0. The .text_vle code section will have ELF section header flags marking it as VLE code, not Book E code.
-Xasm-debug-on	Generate debug line and file information

Table 3-6. Linker Options

Option	Description
-tPPCE200Z3VEN:simple	Selects target processor: PPCE200Z3, generates ELF using EABI conventions, NO floating point support, selects simple environment settings for Startup Module and Libraries.
-Xelf	Generates ELF object format for output file
-m6	Generates a detailed link map and cross reference table
-lc	Specifies to linker to search for libc.a
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files.
-Xlibc-old	Enables usage of legacy (pre-release 5.6) libraries

3.1.3 GHS Compiler/Linker/Assembler Options

Table 3-7. Compiler Options

Option	Description
-cpu=ppc563xm	Selects target processor: ppc563xm
-ansi	Enforces strict ANSI mode (C89 standard)
-noSPE	Disables the use of SPE and vector floating point instructions by the compiler.
-Ospace	Optimize for size
-sda=0	Enables the Small Data Area optimization with a threshold of 0.
--no_commons	Allocates uninitialized global variables to a section and initializes them to zero at program startup. This may improve optimizations by giving the compiler optimizer more information about the location of the variable.
-vle	Enables VLE code generation
-dual_debug	Enables the generation of DWARF, COFF, or BSD debugging information in the object file
-G	Generates source level debugging information and allows procedure call from debugger's command line.
--no_exceptions	Disables support for exception handling

Table continues on the next page...

Table 3-7. Compiler Options (continued)

Option	Description
-Wundef	Generates warnings for undefined symbols in preprocessor expressions
-Wimplicit-int	Issues a warning if the return type of a function is not declared before it is called
-Wshadow	Issues a warning if the declaration of a local variable shadows the declaration of a variable of the same name declared at the global scope, or at an outer scope
-Wtrigraphs	Issues a warning for any use of trigraphs
--prototype_errors	Generates errors when functions referenced or called have no prototype
--incorrect_pragma_warnings	Valid #pragma directives with wrong syntax are treated as warnings
-noslashcomment	C++ like comments will generate a compilation error
-preprocess_assembly_files	Preprocesses assembly files
-nostartfile	Do not use Start files
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DGHS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GHS preprocessor symbol.
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.
-DMCAL_CER_VALIDATION	-D defines a preprocessor symbol for CER Report
-DMCAL_VERSION_CHECK	-D defines enable the cross check between the AutoSar component Version Numbers

Table 3-8. Assembler Options

Option	Description
-cpu=ppc563xm	Selects target processor: ppc563xm

Table 3-9. Linker Options

Option	Description
-cpu=ppc563xm	Selects target processor: ppc563xm
-nostartfiles	Do not use Start files.
-vle	Enables VLE code generation
-linker_warnings	Display linker warnings

3.1.4 CSMC Compiler/Linker/Assembler Options

Table 3-10. Compiler Options

Option	Description
-l	Create listing file; this option directs the compiler to produce an assembly language file with C source line interspersed in it. Please note that the C source lines are commented in the assembly language file: they start with ';'.
+modvc	Memory model with "medium size" application, in detail: "data" less than 64kb, "constants" less than 64kb, no code size limit
+rev	Tells the compiler to reverse the order of bits in the bitfields. You need this option in order to use most non-Cosmic header files.
-pc99	authorize the repetition of the const and volatile modifiers in the declaration either directly or indirectly in the typedef.
-odB5	disable the optimization B5.
-pxf	prefix filenames in the debug information with absolute full path name.
+debug	produce debug information to be used by the debug utilities provided with the compiler and by any external debugger.
-DCSMC	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the CSMC preprocessor symbol.
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.
-DMCAL_CER_VALIDATION	-D defines a preprocessor symbol for CER Report
-DMCAL_VERSION_CHECK	-D defines enable the cross check between the AutoSar component Version Numbers

Table 3-11. Assembler Options

Option	Description
-l	create a listing file. The name of the listing file is derived from the input file name by replacing the suffix by the ".ls" extension

Table 3-12. Linker Options

Option	Description
-p	display symbols with physical address instead of logical address in the map file.

3.2 Files required for Compilation

This section describes the include files required to compile, assemble (if assembler code) and link the MCU driver for MPC5634M microcontrollers.

To avoid integration of incompatible files, all the include files from other modules shall have the same AR_MAJOR_VERSION and AR_MINOR_VERSION, i.e. only files with the same AUTOSAR major and minor versions can be compiled.

MCU Files

- MCU_TS_T2D14M20I0R0\src\Dma_MCU_LLD.c
- MCU_TS_T2D14M20I0R0\src\eMIOS_MCU_LLD.c
- MCU_TS_T2D14M20I0R0\src\Flash_MCU_LLD.c
- MCU_TS_T2D14M20I0R0\src\FMPLL_MCU_LLD.c
- MCU_TS_T2D14M20I0R0\src\MCU.c
- MCU_TS_T2D14M20I0R0\src\MCU_Irq.c
- MCU_TS_T2D14M20I0R0\src\MCU_LLD.c
- MCU_TS_T2D14M20I0R0\src\Reg_eSys_EMIOS.c
- MCU_TS_T2D14M20I0R0\src\SIU_MCU_LLD.c
- MCU_TS_T2D14M20I0R0\include\Dma_LLD.h
- MCU_TS_T2D14M20I0R0\include\eMIOS_MCU_LLD.h
- MCU_TS_T2D14M20I0R0\include\FMPLL_MCU_LLD.h
- MCU_TS_T2D14M20I0R0\include\MCU.h
- MCU_TS_T2D14M20I0R0\include\MCU_LLD.h
- MCU_TS_T2D14M20I0R0\include\Reg_eSys_DMA.h
- MCU_TS_T2D14M20I0R0\include\Reg_eSys_EMIOS.h
- MCU_TS_T2D14M20I0R0\include\Reg_eSys_FLASHC.h
- MCU_TS_T2D14M20I0R0\include\Reg_eSys_FMPLL.h
- MCU_TS_T2D14M20I0R0\include\Reg_eSys_SIU.h
- MCU_TS_T2D14M20I0R0\include\SIU_MCU_LLD.c

MCU Generated Files

- MCU_TS_T2D14M20I0R0\generate_PC\src\MCU_Cfg.c (For PC Variant) - This file should be generated by the user using a configuration tool for compilation
- MCU_TS_T2D14M20I0R0\generate_PC\include\MCU_Cfg.h - This file should be generated by the user using a configuration tool for compilation
- MCU_TS_T2D14M20I0R0\generate_PB\src\MCU_PBcfg.c (For PB Variant) - This file should be generated by the user using a configuration tool for compilation

Files from Base common folder

- Base_TS_T2D14M20I0R0\include\Cer.h
- Base_TS_T2D14M20I0R0\include\Compiler.h
- Base_TS_T2D14M20I0R0\include\Compiler_Cfg.h
- Base_TS_T2D14M20I0R0\include\ComStack_Types.h
- Base_TS_T2D14M20I0R0\include\Mcal.h
- Base_TS_T2D14M20I0R0\include\MemMap.h
- Base_TS_T2D14M20I0R0\include\Platform_Types.h
- Base_TS_T2D14M20I0R0\include\Soc_Ips.h

- Base_TS_T2D14M20I0R0\include\Std_Types.h
- Base_TS_T2D14M20I0R0\generate_PC\include\modules.h

Files from Reg folder:

- Base_TS_T2D14M20I0R0\include\Reg_eSys.h
- Base_TS_T2D14M20I0R0\include\Reg_Macros.h

Files from Dem folder:

- Dem_TS_T2D14M20I0R0\include\Dem.h
- Dem_TS_T2D14M20I0R0\include\Dem.IntErrId.h
- Dem_TS_T2D14M20I0R0\include\Dem_Types.h
- Dem_TS_T2D14M20I0R0\src\Dem.c

Files from Det folder:

- Det_TS_T2D14M20I0R0\include\Det.h
- Det_TS_T2D14M20I0R0\src\Det.c

Files from SchM folder:

- SchM_TS_T2D14M20I0R0\include\SchM_Adc.h
- SchM_TS_T2D14M20I0R0\src\SchM_Adc.c
- SchM_TS_T2D14M20I0R0\include\SchM_Can.h
- SchM_TS_T2D14M20I0R0\src\SchM_Can.c
- SchM_TS_T2D14M20I0R0\include\SchM_Dio.h
- SchM_TS_T2D14M20I0R0\src\SchM_Dio.c
- SchM_TS_T2D14M20I0R0\include\SchM_Fls.h
- SchM_TS_T2D14M20I0R0\src\SchM_Fls.c
- SchM_TS_T2D14M20I0R0\include\SchM_Gpt.h
- SchM_TS_T2D14M20I0R0\src\SchM_Gpt.c
- SchM_TS_T2D14M20I0R0\include\SchM_Icu.h
- SchM_TS_T2D14M20I0R0\src\SchM_Icu.c
- SchM_TS_T2D14M20I0R0\include\SchM_Mcu.h
- SchM_TS_T2D14M20I0R0\src\SchM_Mcu.c
- SchM_TS_T2D14M20I0R0\include\SchM_Port.h
- SchM_TS_T2D14M20I0R0\src\SchM_Port.c
- SchM_TS_T2D14M20I0R0\include\SchM_Pwm.h
- SchM_TS_T2D14M20I0R0\src\SchM_Pwm.c
- SchM_TS_T2D14M20I0R0\include\SchM_RamTst.h
- SchM_TS_T2D14M20I0R0\src\SchM_RamTst.c
- SchM_TS_T2D14M20I0R0\include\SchM_Spi.h
- SchM_TS_T2D14M20I0R0\src\SchM_Spi.c
- SchM_TS_T2D14M20I0R0\include\SchM_Wdg.h
- SchM_TS_T2D14M20I0R0\src\SchM_Wdg.c

3.3 Setting up the Plug-ins

The MCU driver was designed to be configured by using the EB Tresos Studio (version Tresos 2010a.sr4 20100415-release2010a-sr4 or later).

Location of various files inside the MCU module folder:

- VSMD (Vendor Specific Module Definition) file in EB tresos Studio XDM format:
  - ..\ Mcu _ TS_T2D14M20I0R0 \config\Mcu.xdm
- VSMD (Vendor Specific Module Definition) file(s) in AUTOSAR compliant EPD format:
  - ..\ Mcu _ TS_T2D14M20I0R0 \autosar\MCU.epd
- Code Generation Templates for Pre-Compile time configuration parameters:
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PC\src\MCU_Cfg.c
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PC\include\MCU_Cfg.h
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PC\MCU_checkvalues.m
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PC\MCU_RegOperations.m
- Code Generation Templates for Post-Build time configuration parameters:
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PB\src\MCU_PBcfg.c
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PB\MCU_checkvalues.m
  - ..\ Mcu _ TS_T2D14M20I0R0\generate_PB\MCU_RegOperations.m

Steps to generate the configuration:

1. Copy the module folders Base_ TS_T2D14M20I0R0 , Resource_ TS_T2D14M20I0R0 , Dem_ TS_T2D14M20I0R0 , Det_ TS_T2D14M20I0R0, Mcu_ TS_T2D14M20I0R0, SchM_TS_T2D14M20I0R0 into the Tresos plugins folder.
2. Set the desired Tresos Output location folder for the generated sources and header files.
3. Use the EB tresos Studio GUI to modify ECU configuration parameters values.
4. Generate the configuration files.

Chapter 4
Function calls to module

None

4.1 Function Calls during Start-up

The first BSW module to be initialized after Power on shall be MCU. The MCU shall be initialized in the following sequence.

1. Mcu_Init()
2. Mcu_InitClock()
3. Mcu_GetPllStatus () – Till PLL is locked.
4. Mcu_DistributePllClock()
5. Mcu_InitRamSection() – If required

4.2 Function Calls during Shutdown

Mcu_SetMode (sleep mode) API shall be called during GO SLEEP phase of the EcuM's Shutdown state to configure the hardware for Sleep mode. This shall be called after ICU & GPT are set to sleep.

4.3 Function Calls during Wake-up

None

Chapter 5
Module requirements

None

5.1 Exclusive areas to be defined in BSW scheduler

The following functions are used to access critical regions in the MCU driver:

SchM_Enter_Mcu(SCHM_MCU_INSTANCE_0, MCU_EXCLUSIVE_AREA_00);
SchM_Exit_Mcu(SCHM_MCU_INSTANCE_0, MCU_EXCLUSIVE_AREA_00);

SchM_Enter_Mcu(SCHM_MCU_INSTANCE_0, MCU_EXCLUSIVE_AREA_01);
SchM_Exit_Mcu(SCHM_MCU_INSTANCE_0, MCU_EXCLUSIVE_AREA_01);

The following critical regions are used in the MCU driver:

4.1.1 MCU_EXCLUSIVE_AREA_00

Protects the write to the BIUCR0, BIUCR1 and BIUAPR registers performed by the function Mcu_Flash_Init invoked by nested method calls started by the API Mcu_Init.

4.1.2 MCU_EXCLUSIVE_AREA_01

Protects the write to the BIUCR0 and BIUCR1 registers performed by the function Mcu_Flash_Configure invoked by nested method calls started by the API Mcu_InitClock.

5.2 Peripheral Hardware Requirements

Not applicable

5.3 ISR to configure within OS – dependencies

The following ISR's are used by the MCU driver:

Table 5-1. MCU ISRs

ISR Name	Hardware interrupt vector
Mcu_ClockLoss_ISR	43
Mcu_LockLoss_ISR	44

5.4 ISR Macro

MCAL drivers use the ISR macro to define the functions that will process hardware interrupts. Depending on whether the OS is used or not, this macro can have different definitions:

a. OS is not used - AUTOSAR_OS_NOT_USED is defined:

i. If USE_SW_VECTOR_MODE is defined:

```
#define ISR(IsrName) void IsrName(void)
```

In this case, drivers' interrupt handlers are normal C functions and the prolog/epilog handle the context save and restore.

ii. If USE_SW_VECTOR_MODE is not defined:

```
#define ISR(IsrName) INTERRUPT_FUNC void IsrName(void)
```

In this case, drivers' interrupt handlers must save and restore the execution context.

Custom OS is used - AUTOSAR_OS_NOT_USED is not defined

```
#define ISR(IsrName) void OS_isr_##IsrName()
```

In this case, OS is handling the execution context when an interrupt occurs. Drivers' interrupt handlers are normal C functions.

Other vendor's OS is used - AUTOSAR_OS_NOT_USED is not defined. Please refer to the OS documentation for description of the ISR macro.

5.5 Other AUTOSAR modules - dependencies

- **Base**

- **Det:** This module is necessary for enabling Development error detection. The API function used is Det_ReportError(). The activation/deactivation of Development error detection is configurable using 'McuDevErrorDetect' configuration parameter.
- **Dem:** This module is necessary for enabling reporting of production relevant error status. The API function used is Dem_ReportErrorStatus().
- **Schedule Manager:** This module is necessary for enabling and disabling interrupts globally to implement a critical section. The API funzion used are SchM_Enter_Mcu() and SchM_Exit_Mcu().
- **Resource:** Sub-Derivative model is selected from Resource configuration.

Chapter 6
Main API Requirements

6.1 Main functions calls within BSW scheduler

None

6.2 API Requirements

None.

6.3 Calls to Notification Functions, Callbacks, Callouts

None.

Chapter 7
Memory Allocation

7.1 Sections to be defined in MemMap.h

For Post Build data:

```
#ifdef MCU_START_CONFIG_DATA_UNSPECIFIED
#undef MCU_START_CONFIG_DATA_UNSPECIFIED
#undef MEMMAP_ERROR
/*Memory Section for Post Build Data to be defined here. Example given in the next line*/
#pragma ghs section const=".pbMcu_cfg"
#endif
#ifdef MCU_STOP_CONFIG_DATA_UNSPECIFIED
#undef MCU_STOP_CONFIG_DATA_UNSPECIFIED
#undef MEMMAP_ERROR
/*End of section to be mentioned here. Example given in the next line.*/
#pragma ghs section
#endif
```

For Code:

```
#ifdef MCU_START_SEC_CODE
#undef MCU_START_SEC_CODE
#undef MEMMAP_ERROR
/*Memory Section for Code to be defined here.*/
#endif
#ifdef MCU_STOP_SEC_CODE
#undef MCU_STOP_SEC_CODE
#undef MEMMAP_ERROR
/*End of section to be mentioned here*/
#endif
#ifdef MCU_START_SEC_RAMCODE
#undef MCU_START_SEC_RAMCODE
#undef MEMMAP_ERROR
/*Memory Section for Code to be defined here.*/
#endif
#ifdef MCU_STOP_SEC_RAMCODE
#undef MCU_STOP_SEC_RAMCODE
#undef MEMMAP_ERROR
/*End of section to be mentioned here*/
#endif
```

For Variables:

```
#ifdef MCU_START_SEC_VAR_UNSPECIFIED
#undef MCU_START_SEC_VAR_UNSPECIFIED
#undef MEMMAP_ERROR
/*Memory Section for Variables to be defined here.*/
#endif
#ifdef MCU_STOP_SEC_VAR_UNSPECIFIED
#undef MCU_STOP_SEC_VAR_UNSPECIFIED
#undef MEMMAP_ERROR
/*End of section to be mentioned here*/
#endif
```

For Constant data:

```
#ifdef MCU_START_SEC_CONST_UNSPECIFIED
#undef MCU_START_SEC_CONST_UNSPECIFIED
#undef MEMMAP_ERROR
/*Memory Section for Constants to be defined here.*/
#endif
#ifdef MCU_STOP_SEC_CONST_UNSPECIFIED
#undef MCU_STOP_SEC_CONST_UNSPECIFIED
#undef MEMMAP_ERROR
/*End of section to be mentioned here*/
#endif
```

7.2 Linker command file

Memory shall be allocated for every section defined in MemMap.h

MCU_START_SEC_RAMCODE and MCU_STOP_SEC_RAMCODE are used to let the linker allocate the code found between these two defines into the RAM. They have been implemented in MemMap.h, which defines a new pragma section ".ramcode", so in the linker command file the ".ramcode" section should be defined; then a separate routine should be implemented by the user to copy this section from flash to ram, for example into startup file.

In the case of MCU driver, the functions Mcu_Flash_Init_Ram and Mcu_Flash_Configure_Ram use these defines in order to be executed from RAM during FLASH memory configurations performed by themselves.

Chapter 8
Configuration parameters considerations

Configuration parameter class for Autosar MCU driver fall into the following variants as defined below:

8.1 Configuration Parameters

Specifies whether the configuration parameter shall be of configuration class Post Build

Table 8-1. Configuration Parameters

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
McuGeneral	McuDevErrorDetect	Pre Compile parameter for all Variants of configuration	Pre Compile
McuGeneral	McuPerformResetApi	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuGeneral	McuCalloutBeforePerformReset	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuGeneral	McuPerformResetCallout	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuGeneral	McuVersionInfoApi	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuGeneral	McuConfigureFlash	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuGeneral	McuEnterLowPowerMode	Pre Compile parameter for all Variants of Configuration	Pre Compile
McuModuleConfiguration	McuClockSrcFailureNotification	VariantPC or VariantPB	Post Build
McuModuleConfiguration	McuNumberOfMcuModes	VariantPC or VariantPB	Post Build
McuModuleConfiguration	McuRamSectors	VariantPC or VariantPB	Post Build
McuModuleConfiguration	McuResetSetting	VariantPC or VariantPB	Post Build
McuClockSettingConfig	CrystalFrequencyHz	VariantPC or VariantPB	Post Build
McuClockSettingConfig	McuTimeout	VariantPC or VariantPB	Post Build

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
McuClockReferencePoint	McuClockReferencePointFrequency	VariantPC or VariantPB	Post Build
GeneralClockSettings	McuOperatingMode	VariantPC or VariantPB	Post Build
GeneralClockSettings	McuBypassDivider	VariantPC or VariantPB	Post Build
GeneralClockSettings	McuSystemClockDivider	VariantPC or VariantPB	Post Build
McuExternalClock	McuExternalBusTapSelect	VariantPC or VariantPB	Post Build
McuExternalClock	McuExternalBusDivisonFactor	VariantPC or VariantPB	Post Build
McuPll	McuPllClockFrequency	VariantPC or VariantPB	Post Build
McuPll	McuPllMode	VariantPC or VariantPB	Post Build
McuPll	McuPllClockReference	VariantPC or VariantPB	Post Build
McuPllParameter	McuFeedbackDivideRatio	VariantPC or VariantPB	Post Build
McuPllParameter	McuInputDivideRatio	VariantPC or VariantPB	Post Build
McuPllParameter	McuOutputDivideRatio	VariantPC or VariantPB	Post Build
McuPllParameter	MODEN	VariantPC or VariantPB	Post Build
McuPllParameter	MODSEL	VariantPC or VariantPB	Post Build
McuPllParameter	McuModulationRate	VariantPC or VariantPB	Post Build
McuPllParameter	McuPeakToPeakModulationDepth	VariantPC or VariantPB	Post Build
McuPllParameter	McuLossOfClockEnable	VariantPC or VariantPB	Post Build
McuPllParameter	McuLossOfLockResetEnable	VariantPC or VariantPB	Post Build
McuPllParameter	McuLossOfClockResetEnable	VariantPC or VariantPB	Post Build
McuPllParameter	McuLossOfLockInterruptRequest	VariantPC or VariantPB	Post Build
McuPllParameter	McuLossOfClockInterruptRequest	VariantPC or VariantPB	Post Build
McuEMIOSSettings	GlobalPrescaler	VariantPC or VariantPB	Post Build
McuEMIOSSettings	ServerTimeSlot	VariantPC or VariantPB	Post Build
McuEMIOSSettings	ExternalTimeBase	VariantPC or VariantPB	Post Build
McuEMIOSSettings	MdisBit	VariantPC or VariantPB	Post Build
McuEMIOSSettings	FreezeBit	VariantPC or VariantPB	Post Build
McuEMIOSSettings	GlobalTimeBaseEnable	VariantPC or VariantPB	Post Build
McuEMIOSSettings	GlobalPrescalerEnable	VariantPC or VariantPB	Post Build
McuFlashBIUCR0	AddressPipeliningControl	VariantPC or VariantPB	Post Build
McuFlashBIUCR0	WriteWaitStateControl	VariantPC or VariantPB	Post Build
McuFlashBIUCR0	ReadWaitStateControl	VariantPC or VariantPB	Post Build

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
McuInterrupt	McuClockLoss	VariantPC or VariantPB	Post Build
McuInterrupt	McuLockLoss	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority0	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority4	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority8	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority12	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority16	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority20	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority24	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	eDMAChannelPriority28	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	ERGA	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	ERCA	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	EDBG	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	EBW	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	GRP0PRI	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	GRP1PRI	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	GRP2PRI	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	GRP3PRI	VariantPC or VariantPB	Post Build
McuEDMA_A_Config	DmaTimeOut	VariantPC or VariantPB	Post Build
McuFlashBIUCR	Master 0 Prefetch Enable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	Master 1 Prefetch Enable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	Master 2 Prefetch Enable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	Master 3 Prefetch Enable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	McuDataPrefetchEnable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	McuInstructionPrefetchEnable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	McuPFLASHPrefetchLimit	VariantPC or VariantPB	Post Build
McuFlashBIUCR	McuPFLASHLineReadBuffersEnable	VariantPC or VariantPB	Post Build
McuFlashBIUCR	GlobalConfigurationEnable	VariantPC or VariantPB	Post Build
McuFlashBIUAPR	Master 0 Access Protection	VariantPC or VariantPB	Post Build
McuFlashBIUAPR	Master 1 Access Protection	VariantPC or VariantPB	Post Build
McuFlashBIUAPR	Master 2 Access Protection	VariantPC or VariantPB	Post Build
McuFlashBIUAPR	Master 3 Access Protection	VariantPC or VariantPB	Post Build
McuFlashBIUCR2	McuLineBufferConfiguration	VariantPC or VariantPB	Post Build
McuResetSource	McuResetType	VariantPC or VariantPB	Post Build

Table continues on the next page...

Table 8-1. Configuration Parameters (continued)

Configuration Container	Configuration Parameters	Configuration Variant	Current Implementation
McuModeSettingConf	McuMode	VariantPC or VariantPB	Post Build
HaltRegister	SIU_HLTACKTimeOut	VariantPC or VariantPB	Post Build
HaltRegister	HaltCPU	VariantPC or VariantPB	Post Build
HaltRegister	HaltNDEDI	VariantPC or VariantPB	Post Build
HaltRegister	HalteTPU	VariantPC or VariantPB	Post Build
HaltRegister	HaltNPC	VariantPC or VariantPB	Post Build
HaltRegister	HaltEBI	VariantPC or VariantPB	Post Build
HaltRegister	HaltADC	VariantPC or VariantPB	Post Build
HaltRegister	HalteMIOS	VariantPC or VariantPB	Post Build
HaltRegister	HaltDFILT	VariantPC or VariantPB	Post Build
HaltRegister	HaltFlexCAN_C	VariantPC or VariantPB	Post Build
HaltRegister	HaltFlexCAN_A	VariantPC or VariantPB	Post Build
HaltRegister	HaltDSPI_C	VariantPC or VariantPB	Post Build
HaltRegister	HaltDSPI_B	VariantPC or VariantPB	Post Build
HaltRegister	HalteSCI_B	VariantPC or VariantPB	Post Build
HaltRegister	HalteSCI_A	VariantPC or VariantPB	Post Build
McuRamSectorSettingConf	McuRamDefaultValue	VariantPC or VariantPB	Post Build
McuRamSectorSettingConf	McuRamSectionBaseAddress	VariantPC or VariantPB	Post Build
McuRamSectorSettingConf	McuRamSectionSize	VariantPC or VariantPB	Post Build
McuRamSectorSettingConf	McuRamSectionBaseAddrLinkerSym	VariantPC or VariantPB	Post Build
McuRamSectorSettingConf	McuRamSectionSizeLinkerSym	VariantPC or VariantPB	Post Build

Chapter 9
Integration Steps

This section gives a brief overview of the steps needed for integrating MCU Driver :

- Generate the required MCU configurations. For more details refer to Setting up the Plug-ins
- Allocate proper memory sections in MemMap.h and linker command file. For more details refer to section Sections to be defined in MemMap.h
- Make sure all include files for compilation. For more details refer to section Files required for Compilation
- Map the ISRs to their vector locations. For more details refer to section ISR to configure within OS – dependencies
- Compile and build the MCU with all the dependent modules. For more details refer to section Building the Driver

Chapter 10
ISR Reference

ISR functions exported by the MCU driver.

10.1 Software specification

The following sections contains driver software specifications.

10.1.1 Define Reference

Constants supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

10.1.2 Enum Reference

Enumeration of all constants supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

10.1.3 Function Reference

Functions of all functions supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

10.1.3.1 Function Mcu_ClockLoss_ISR

This function represents the ISR for loss of clock.

Details:

If the loss of clock interrupt is enabled and an invalid condition is detected, then the interrupt is triggered and the event is reported.

Note

Violates MISRA 2004 Advisory Rule 19.1, 19.15 : See `Mcu_IRQ_c_REF_1`, Mcu_IRQ_c_REF_5 above

Note

Violates MISRA 2004 Advisory Rule 8.1, See `Mcu_IRQ_c_REF_4` above

Prototype: `void Mcu_ClockLoss_ISR(void);`

10.1.3.2 Function Mcu_LockLoss_ISR

This function represents the ISR for loss of lock.

Details:

If the loss of lock interrupt is enabled and an invalid condition is detected, then the interrupt is triggered and the event is reported.

Note

Violates MISRA 2004 Advisory Rule 8.1, See `Mcu_IRQ_c_REF_4`

Prototype: `void Mcu_LockLoss_ISR(void);`

10.1.4 Structs Reference

Data structures supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

10.1.5 Types Reference

Types supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

10.1.6 Variables Reference

Variables supported by the driver are as per AUTOSAR MCU Driver software specification Version 3.0 .

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: IM14MCUASR3.0R2.0.0
Rev. 1.1