Release Notes

for S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2

Document Number: Release Notes for S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2
Rev. 1.0

Contents

Section number **Title** **Page**

Chapter 1
Getting Started

Chapter 2
Release Specifics

Chapter 3
Known Issues for S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2

Chapter 4
Changes List for S32K14X AUTOSAR 4.2 SMCAL

Chapter 1
Getting Started

1.1 Package content

This package contains the NXP S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2:

- "eclipse/plugins/<mod>_TS_T40D2M10I2R0" directories - Tresos Plugins, 1 per module.
- "S32K14X_SMCAL4.2_RTM_1.0.2_Sample_Application" - Folder containing the SMCAL Sample Application.
- "S32K14X_SMCAL4.2_RTM_1.0.2_ReleaseNotes.pdf" - This file.
- "S32K14X_SMCAL4.2_Safety_Manual.pdf" - Safety Manual.
- Various other files: GettingStarted.htm start page and associated images, the license.txt EULA file and the Uninstall.exe utility for removing the SMCAL installation.

1.2 Installation

Follow the installer steps. By default the installer will create a link between the installation target directory and a selected EB Tresos installation. If you choose not to create a link, you can later create one manually or you can copy all "<mod>TS_T40D2M10I2R0" directories and .JAR files to the "<Tresos Install Path>\plugins" directory.

Chapter 2
Release Specifics

The S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2 is AUTOSAR 4.2.2 compliant. The AUTOSAR Configuration ARXML specification takes precedence over AUTOSAR SWS PDF Specifications if there are discrepancies.

2.1 Release Details

This is the AUTOSAR 4.2 SMCAL RTM 1.0.2 release for the S32K14X platform.

The Sample Application included in this release contains some basic examples of usage for the SMCAL drivers. It also includes an example of integration of the SMCAL drivers with the NXP AUTOSAR OS/S32K v4.0 EAR v0.8.0 (the OS installer comes as a different NXP software package).

This release has RTM quality status in terms of testing and quality documentation.

This release contains a deviation from AUTOSAR recommended version check inside source files. In all source files, Software Version values are checked (major, minor, patch). AUTOSAR release or SWS versions are not checked during preprocessing/ template generation. The correct SWS versions are exported by each module.

The deviations from AUTOSAR are described in the User's Manual of each SMCAL driver (*Deviation from Requirements* chapter).

This release was developed and tested using:

- Motherboard S32K-MB
- S32K144/S32K142 MINI-MODULE S32K144-100LQFP
- S32K148/S32K146 MINI-MODULE S32K14xCVD-Q144
- S32K118 MINI-MODULE S32K14XCVD-Q064
- S32K142 DEVICE 100LQFP PS32K142UAVLLT 0N33V

- S32K144 DEVICE 100LQFP PS32K144UAVLLA 0N57U
- S32K146 DEVICE 144LQFP PS32K146UAVLQ 0N73V
- S32K148 DEVICE 144LQFP PS32K148UAVLQ 0N20V
- S32K118 DEVICE 64LQFP PS32K118LAMLH 0N97V

The functions contained in the CanIf, DEM, DET, EcuC, EcuM, EthIf, EthTrcv, LinIf, MemIf, Os, RTE, WdgIf plugins are sample stub functions. These functions should be replaced by the user developed code during integration.

The Resource module is needed to select the MCU derivative. The derivatives supported can be found in the Resource module definition file, parameter 'ResourceSubderivative'.

2.2 Used Documentation

This release was developed and tested with the following documents:

Table 2-1. Reference Manuals

Document Title	Version and Date
S32K14X Reference Manual	Reference Manual, Rev. 9, 9/2018

Table 2-2. Implemented Errata

Document Title	Maskset	Date
S32K142 Mask Set Errata for Mask 0N33V	0N33V	30/11/2017
S32K144 Mask Set Errata for Mask 0N57U	0N57U	30/11/2017
S32K146 Mask Set Errata for Mask 0N73V	0N73V	30/11/2017
S32K148 Mask Set Errata for Mask 0N20V	0N20V	25/10/2018
S32K118 Mask Set Errata for Mask 0N97V	0N97V	07/01/2019

2.3 Supported Derivatives

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

- s32k148_lqfp144

- s32k148_lqfp176
- s32k148_mapbga100
- s32k146_lqfp144
- s32k146_lqfp100
- s32k146_lqfp64
- s32k146_mapbga100
- s32k144_lqfp100
- s32k144_lqfp64
- s32k144_mapbga100
- s32k142_lqfp100
- s32k142_lqfp64
- s32k118_lqfp48
- s32k118_lqfp64
- s32k142_lqfp48
- s32k144_lqfp48
- s32k148_lqfp100

2.4 Modules Configuration

Modules configurations were developed and tested using the Tresos Configuration Tool version "*EB tresos Studio 23.0.0 b170330-0431*"

Configuration definition files were developed according to AUTOSAR 4.2.2, AUTOSAR_EcucParamDef.arxml

A folder named "<mod>_TS_TtDdMmIiRr" exists for each delivered module (<mod>). It is called a Tresos plugin for the module. A plugin contains the AUTOSAR module definition file (epd), the Tresos Xpath Data Model module definition file (xdm), the module user and integration manuals, the module configuration generation template source files, and the module driver static source files. Additional necessary Tresos specific tooling files are also included.

Plugin Encoding: <mod>_TS_TtDdMmIiRr

Important change related to the plugin notation:
- "m" = coding major and minor version number, can contain 1 or more digits
- "i" = patch number.
The major version number will be left out, if it is "0", in this case "m" contains 1 digit only,
otherwise it contains 2 digits

For this release:

- t=40, CortexM Architecture
- d=2, S32K14X (derivative)
- m=10, Release major and minor version
- i=2, Release patch version
- r=0, Reserved

2.5 Support and Driver Plugins Delivered

Table 2-3. Support and Driver Plugins Delivered

Plugin	SW Version	Description
ADC	sw version 1.0.2	Driver, Analog to Digital Conversion
Base	sw version 1.0.2	Base Module, General AUTOSAR and Hardware Specific register files
CAN	sw version 1.0.2	Driver, Controller Area Network
CanIf	sw version 1.0.2	Support Stub, Controller Area Network Interface
CRCU	sw version 1.0.2	Driver, Cyclic Redundancy Check Unit
DEM	sw version 1.0.2	Support Stub, Diagnostic Event Manager
DET	sw version 1.0.2	Support Stub, Development Error Tracer
DIO	sw version 1.0.2	Driver, Digital Input Output
EcuC	sw version 1.0.2	Support Stub, ECU Configuration
EcuM	sw version 1.0.2	Support Stub, ECU State Manager
EEP	sw version 1.0.2	Driver, EEPROM
ETH	sw version 1.0.2	Driver, Ethernet
EthIf	sw version 1.0.2	Support Stub, Ethernet Interface
EthTrcv	sw version 1.0.2	Driver, Ethernet Transceiver
FEE	sw version 1.0.2	Driver, Flash EEPROM Emulation
FLS	sw version 1.0.2	Driver, Flash
GPT	sw version 1.0.2	Driver, General Purpose Timer
I2C	sw version 1.0.2	Driver, Inter-Integrated Circuit
ICU	sw version 1.0.2	Driver, Input Capture Unit

Table continues on the next page...

Table 2-3. Support and Driver Plugins Delivered (continued)

Plugin	SW Version	Description
LIN	sw version 1.0.2	Driver, Local Interconnect Network
LinIf	sw version 1.0.2	Support Stub, Local Interconnect Network Interface
MCEM	sw version 1.0.2	Driver, Microcontroller Error Manager
MCL	sw version 1.0.2	Driver, Microcontroller Library
MCU	sw version 1.0.2	Driver, Microcontroller Unit
MemIf	sw version 1.0.2	Support Stub, Memory Interface
OCU	sw version 1.0.2	Driver, Output Control Unit
OS	sw version 1.0.2	Support stub, Operating System
PORT	sw version 1.0.2	Driver, Port
PWM	sw version 1.0.2	Driver, Pulse Width Modulation
Resource	sw version 1.0.2	Resource Module, Required by all other modules to select MCU derivative
RTE	sw version 1.0.2	Support Stub, only for Schedule Manager
SPI	sw version 1.0.2	Driver, Serial Peripheral Interface
WDG	sw version 1.0.2	Driver, Watchdog
WdgIf	sw version 1.0.2	Support Stub, Watchdog Interface

2.6 Module Plugin Folder Structure

Table 2-4. Module Plugin Folder Structure

Folder or file	Description
<mod>_TS_TtDdMmIiRr\anchors.xml	Tresos Configuration tooling documentation data file
<mod>_TS_TtDdMmIiRr\plugin.xml	Tresos Configuration tooling data file
<mod>_TS_TtDdMmIiRr\autosar\<mod>.epd	Module Parameter Definition in AUTOSAR format
<mod>_TS_TtDdMmIiRr\config\<mod>.xdm	Module Parameter Definition in Tresos XDM format
<mod>_TS_TtDdMmIiRr\config_ext\<mod>PreConfiguration.xdm	Module Parameter Default Configuration in Tresos XDM format[1]
<mod>_TS_TtDdMmIiRr\doc\AUTOSAR_SMCAL_<mod>_IM.pdf	Module Integration Manual
<mod>_TS_TtDdMmIiRr\doc\AUTOSAR_SMCAL_<mod>_UM.pdf	Module User's Manual
<mod>_TS_TtDdMmIiRr\generate_PB	Post-build source files macros
<mod>_TS_TtDdMmIiRr\generate_PB\src	Post-build source file templates
<mod>_TS_TtDdMmIiRr\generate_LT\src	Link-time source file templates (only if applicable)
<mod>_TS_TtDdMmIiRr\generate_PC\	Pre-compile source files macros
<mod>_TS_TtDdMmIiRr\generate_PC\src	Pre-compile source files templates
<mod>_TS_TtDdMmIiRr\generate_swcd	Module BSWMD file
<mod>_TS_TtDdMmIiRr\include\	Module driver header files
<mod>_TS_TtDdMmIiRr\META-INF	Tresos Configuration tooling data and signature files
<mod>_TS_TtDdMmIiRr\src\	Module driver source files[2]

Notes:

[1] Not available for all plugins.

[2] The Support Stub Resource contains the "resource" folder instead of the "src" folder.

2.7 MCAL Sample Application Folder Structure

Table 2-5. MCAL Sample Application Folder Structure

Folder or file	Description
- bin folder	generated object files and linker output files are stored into this folder
- cfg folder	contains configuration files generated by Tresos tool
- include subfolder	contains files with pre-compile configurations
- src subfolder	contains files with post-build and link-time configurations
- doc folder	contains documentation
- include folder	contains header files for device and types definitions
- make folder	makefiles used for building the application
- src folder	contains the application source code file
- toolchains folder	files needed to build with various toolchains (startup, linker command files)
- makefile file	the MCAL sample application makefile
- makefile_os file	the MCAL and OS sample application makefile
- Modules file	specifies which modules are compiled and linked
- make.bat file	launches the make command
- launch.bat file	contains path to the Tresos Studio installation and launches the make.bat file
- Tresos folder/workspace	contains the Tresos project with the application configuration

2.8 Compiler Options

This release was developed and tested with:

- IAR: V8.11.2
- Green Hills Multi 7.1.4 / Compiler 2017.1.4
- (Linaro GCC 6.3-2017.06~dev) 6.3.1 20170509 (Wed Jan 24 16:21:45 CST 2018 build.sh rev=g27a1317 s=L631 Earmv7 -V release_g27a1317_build_Fed_Earmv7)

2.8.1 IAR Compiler/Linker/Assembler Options

Table 2-6. Compiler Options

Option	Description
--cpu=Cortex-M4	Selects target processor: Arm Cortex M4
--cpu=Cortex-M0+	Selects target processor: Arm Cortex M0+
--cpu_mode=thumb	Selects generating code that executes in Thumb state.
--endian=little	Specifies the endianess of core: little endian.
-Ohz	Sets the optimization level to High, favoring size.
-c	Produces an object file (called input-file.o) for each source file.
--no_clustering	Disables static clustering optimizations.
--no_mem_idioms	Makes the compiler to not optimize code sequences that clear, set, or copy a memory region.
--no_explicit_zero_opt	Places the zero initialized variables in data section instead of bss.
--debug	Makes the compiler include information in the object modules.
--diag_suppress=Pa050	Suppresses diagnostic messages (warnings) about non-standard line endings.
-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
-DIAR	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the IAR preprocessor symbol.
--require_prototypes	Forces the compiler to verify that all functions have proper prototypes.
--no_wrap_diagnostics	Disables line wrapping of diagnostic messages issued by compiler.
--no_system_include	Disables the automatic search for system include files.
-e	Enables language extensions. This option is needed by FLS driver which uses _packed structures.

Table 2-7. Assembler Options

Option	Description
--cpu=Cortex-M4	Selects target processor: Arm Cortex M4
--cpu=Cortex-M0+	Selects target processor: Arm Cortex M0+
--cpu_mode=thumb	Selects generating code that executes in Thumb state.
-g	Use this option to disable the automatic search for system include files.

Table 2-8. Linker Options

Option	Description
--cpu=Cortex-M4	Selects target processor: Arm Cortex M4
--cpu=Cortex-M0+	Selects target processor: Arm Cortex M0+
--map filename	Produces a map file.
--no_library_search	Disables automatic runtime library search.
--entry _start	Treats the symbol _start as a root symbol and as the start of the application.

Table continues on the next page...

Table 2-8. Linker Options (continued)

Option	Description
--enable_stack_usage	Enables stack usage analysis.
--skip_dynamic_initialization	Suppress dynamic initialization during system startup.
--no_wrap_diagnostics	Disables line wrapping of diagnostic messages issued by linker.
--config	Specifies the configuration file to be used by the linker.

2.8.2 GHS Compiler/Linker/Assembler Options

Table 2-9. Compiler Options

Option	Description
-cpu=cortexm4	Selects target processor: Arm Cortex M4
-cpu=cortexm0plus	Selects target processor: Arm Cortex M0+
-ansi	Specifies ANSI C with extensions. This mode extends the ANSI X3.159-1989 standard with certain useful and compatible constructs.
-Osize	Optimize for size.
-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
-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
-Wall	Enables all the warnings about constructions that some users consider questionable, and that are easy to avoid even in conjunction with macros.
--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
--short_enum	Store enumerations in the smallest possible type
-c	Produces an object file (called input-file.o) for each source file.
--no_commons	Allocates uninitialized global variables to a section and initializes them to zero at program startup.
-keeptempfiles	Prevents the deletion of temporary files after they are used. If an assembly language file is created by the compiler, this option will place it in the current directory instead of the temporary directory. Produces an object file (called input-file.o) for each source file.
-list	Creates a listing by using the name of the object file with the .lst extension. Assembler option

Table continues on the next page...

Table 2-9. Compiler Options (continued)

Option	Description
-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
-DDISABLE_MCAL_INTERMODULE_ASR_CHECK	-D defines a preprocessor symbol to disable the inter-module version check for AR_RELEASE versions. DISABLE_MCAL_INTERMODULE_ASR_CHECK: By default in the package, drivers are compiled to perform the inter-module version check as per Autosar BSW004. When the inter-module version check needs to be disabled then the DISABLE_MCAL_INTERMODULE_ASR_CHECK global define must be added to the list of compiler options.
-DGHS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GHS preprocessor symbol.

Table 2-10. Assembler Options

Option	Description
-cpu=cortexm4	Selects target processor: Arm Cortex M4
-cpu=cortexm0plus	Selects target processor: Arm Cortex M0+
-c	Produces an object file (called input-file.o) for each source file.
-preprocess_assembly_files	Preprocesses assembly files
-asm=list	Creates a listing by using the name of the object file with the .lst extension. Assembler option

Table 2-11. Linker Options

Option	Description
-Mn	Map file numeric ordering
-delete	Removal from the executable of functions that are unused and unreferenced
-v	Display removed unused functions
-ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete.
-map	Creates a detailed map file
-keepmap	Keep the map file in the event of a link error
-lstartup	Link libstartup library -Run-time environment startup routines
-lsys	Link libsys library -Run-time environment system routines
-larch	Link libarch library -Target-specific run-time support. Any file produced by the Green Hills Compiler may depend on symbols in this library.
-lansi	Link libansi library -the standard C library
-L(/lib/thumb2)	Link thumb2 library
-lutf8_s32	Include utf8_s32.a to use the Wide Character Functions

2.8.3 GCC Compiler/Linker/Assembler Options

Table 2-12. Compiler Options

Option	Description
-c	Produces an object file (called input-file.o) for each source file.
-Os	Use optimization for size.
-ggdb3	Produce debugging information for use by GDB. Level 3 includes extra information, such as all the macro definitions present in the program.
-mcpu=cortex-m4	Selects target processor: Arm Cortex M4
-mcpu=cortex-m0plus	Selects target processor: Arm Cortex M0+
-mthumb	Selects generating code that executes in Thumb state.
-ansi	Specifies ANSI C with extensions.
-mlittle-endian	Generate code for a processor running in little-endian mode.
-fomit-frame-pointer	Removes the frame pointer for all functions, which might make debugging harder.
-msoft-float	Use software floating-point instructions.
-fno-common	Specifies that the compiler should place uninitialized global variables in the data section of the object file, rather than generating them as common blocks.
-Wall	Enables all the warnings about constructions that some users consider questionable, and that are easy to avoid even in conjunction with macros.
-Wextra	Enables some extra warning flags that are not enabled by '-Wall'.
-Wstrict-prototypes	Warn if a function is declared or defined without specifying the argument types.
-Wno-sign-compare	Do not warn when a comparison between signed and unsigned values could produce an incorrect result when the signed value is converted to unsigned.
-fstack-usage	Geneates an extra file that specifies the maximum amount of stack used, on a per-function basis.
-fdump-ipa-all	Enables all inter-procedural analysis dumps.
-Werror=implicit-function-declaration	Generates an error when the prototype of the function is not defined..
-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
-DGCC	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GCC preprocessor symbol.

Table 2-13. Assembler Options

Option	Description
-mcpu=cortex-m4	Selects target processor: Arm Cortex M4
-mcpu=cortex-m0plus	Selects target processor: Arm Cortex M0+
-c	Produces an object file (called input-file.o) for each source file.
-mthumb	This option specifies that the assembler should start assembling Thumb instructions.
-x assembler-with-cpp	Indicates that the assembly code contains C directives and the C preprocessor must be run.

Table 2-14. Linker Options

Option	Description
-Map=filename	Print a link map to the file mapfile.
-T scriptfile	Use scriptfile as the linker script. This script replaces ld’s default linker script(rather than adding to it), so commandfile must specify everything necessary to describe the output file.
--disable-newlib-supplied-syscalls -specs=nosys.specs	These options support for using newlib on core M0+
-u _printf_float -u _scanf_float	These options support generating profile report.
-nostartfiles	Do not use the standard system startup files when linking
-e _start	Specify that the program entry point is _start
-static	The --static flag tells the linker to link a static, not a dynamically linked
-lc	The -lc flag tells the linker to link this binary against the C library, which is newlib in our case.
-lnosys	The -lnosys flag tells the linker to link this binary against the "nosys" library
$(TOOLCHAIN_DIR)/arm-none-eabi/newlib/lib/thumb/v6-m $(TOOLCHAIN_DIR)/lib/gcc/arm-none-eabi/6.3.1/thumb/v6-m	Library for core M0+
$(TOOLCHAIN_DIR)/arm-none-eabi/newlib/lib/thumb $(TOOLCHAIN_DIR)/arm-none-eabi/newlib/lib)	Library for core M4

Chapter 3
Known Issues for S32K14X AUTOSAR 4.2 SMCAL RTM 1.0.2

3.1 Known Issues

ID	Headline
MCAL-21223	[WDG] Wrong configuration is generated when there are multiple WdgSettingsConfig and there is no Variants supported

Chapter 4
Changes List for S32K14X AUTOSAR 4.2 SMCAL

4.1 RTM 1.0.2

ID	Subtype	Headline and Description
MCAL-21283	New	New Feature [SPI] Implementation of LPSPIx_CFGR1[PINCFG] in MCAL driver ,,"As customer Sensata was asking if LPSPIx_CFGR1[PINCFG] has been supported and we, customer engineers, think that it is good to have this feature in MCAL SPI driver."

MCAL-19632	Bug	[ETH] Eth_Enet_Transmit slowdown performance on A53<*>
		Detailed description (how to reproduce it):
		Change
		u32TimeOut = 1000U;
		do
		Unknown macro: \{ / violates ref Eth_Enet_c_REF_2 MISRA rule 11.3 violates ref Eth_Enet_c_REF_7 MISRA rule 11.1 / REG_WRITE32(Eth_u32BaseAddr[u8CtrlIdx] ENET_TDAR_ADDR16, ENET_TDAR_X_DES_ACTIVE_U32); u32TimeOut ; }
		/ violates ref Eth_Enet_c_REF_2 MISRA rule 11.3 /
		/ violates ref Eth_Enet_c_REF_7 MISRA rule 11.1 /
		/ violates ref Eth_Enet_c_REF_15 MISRA rule 12.4 /
		while((ENET_TDAR_X_DES_ACTIVE_U32 == (REG_READ32(Eth_u32BaseAddr[u8CtrlIdx] ENET_TDAR_ADDR16) & ENET_TDAR_X_DES_ACTIVE_U32)) && (u32TimeOut > 0U));
		To
		REG_WRITE32(Eth_u32BaseAddr[u8CtrlIdx] ENET_TDAR_ADDR16, ENET_TDAR_X_DES_ACTIVE_U32);
		Run test case Eth_TS_int_11 on A53*. This test case pass and fail randomize. When I debug I found that sometime it can not send buffer when we set TDAR bit. if we try to put TDAR again buffer is sent and everything work fine. Don't know why hardware driver isn't send current buffer and no error relate*. Current solution is applied to set TDAR several times but I think it make performance slow down.
		To solve slow down performance:

Table continues on the next page...

ID	Subtype	Headline and Description
		I suggest to check package count (ENET_RMON_TPACKET_ADDR16), I found that package count isn't changed if driver cannot send buffer. Due to if occurs on A53 only and Eth_Enet.c is common file for other platforms. We should have option on EB tresos to turn on/off this workaround solution to make sure that core M4 and other platform we call TDAR only one time. Need get help from hardware design team to understand problem clearly. I and Thong guest that buffer (that we prepared) isn't reflect to hardware immediately and we call TDAR before it has correct buffer. Preconditions: Module: Eth Test: Eth_TS_int_11 Core: A53 Test Case ID (internal TC that caught the defect) optional: Eth_TS_int_11 *_O_*_*bserved behavior: N/A Expected behavior: N/A Note:* in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-20137	New	New Feature [FEE] Update AUTOSAR module SWS specification version ,,"Detailed description (how to reproduce it) We have some defines from make file of each module. For ex: Make file of LIN ### # AUTOSAR module SWS specification version (see 'Document Version' in module's SWS) ### SWS_SPEC_VERSION_MAJOR = 1 SWS_SPEC_VERSION_MINOR = 5 SWS_SPEC_VERSION_PATCH = 0 They will appear in plugin.xml as specVersionMajor=""1"" specVersionMinor=""5"" specVersionPatch=""0"" in tag module This information is ""Document version"" and ""Document version"" isn't available from AutoSAR 4.2.2. Therefore, to make consistence data we should: For AutoSAR 4.0.3: Keep use document version from AutoSAR specification. For AutoSAR 4.2.2: Use* SWS_SPEC_VERSION_MAJOR = 4 SWS_SPEC_VERSION_MINOR = 2 SWS_SPEC_VERSION_PATCH = 2 SWS_SPEC_VERSION_SUFFIX = Rev_0002

Table continues on the next page...

ID	Subtype	Headline and Description

		For AutoSAR 4.3.1: Use*
		SWS_SPEC_VERSION_MAJOR = 4
		SWS_SPEC_VERSION_MINOR = 3
		SWS_SPEC_VERSION_PATCH = 1
		SWS_SPEC_VERSION_SUFFIX = Rev_0001"

MCAL-20142	New	New Feature [EEP] Update AUTOSAR module SWS specification version ,,"Detailed description (how to reproduce it) We have some defines from make file of each module. For ex: Make file of LIN ### ################# # AUTOSAR module SWS specification version (see 'Document Version' in module's SWS) ### ################# SWS_SPEC_VERSION_MAJOR = 1 SWS_SPEC_VERSION_MINOR = 5 SWS_SPEC_VERSION_PATCH = 0 They will appear in plugin.xml as specVersionMajor=""1"" specVersionMinor=""5"" specVersionPatch=""0"" in tag module This information is ""Document version"" and ""Document version"" isn't available from AutoSAR 4.2.2. Therefore, to make consistence data we should: For AutoSAR 4.0.3: Keep use document version from AutoSAR specification. For AutoSAR 4.2.2: Use* SWS_SPEC_VERSION_MAJOR = 4 SWS_SPEC_VERSION_MINOR = 2 SWS_SPEC_VERSION_PATCH = 2 SWS_SPEC_VERSION_SUFFIX = Rev_0002 For AutoSAR 4.3.1: Use* SWS_SPEC_VERSION_MAJOR = 4 SWS_SPEC_VERSION_MINOR = 3 SWS_SPEC_VERSION_PATCH = 1 SWS_SPEC_VERSION_SUFFIX = Rev_0001"
MCAL-20508	New	New Feature [MCU] Support SRAM retention ,,"Detailed description (how to reproduce it): Support to to retain and access SRAM contents across functional resets. The description for the implementation is describe in ""SRAM retention: power modes and resets"" chapter.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: NA Test Case ID (internal TC that caught the defect) optional NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add function support for SRAM retention"
MCAL-20965	New	New Feature [EEP] Update IM to link build option information to gcc.xml instead of linaro.xml ,,"NewWorkDescription: For some recent releases, we are changing Compiler from Linaro > GCC but we are still keeping build option of Linaro in some documents like IM, Release note... Please update: <topicref href=""topics/sub_building_driver/sub_build_options/linaro.xml""/> To: <topicref href=""topics/sub_building_driver/sub_build_options/gcc.xml""/> In: AUTOSAR_MCAL_<MDL>IM.ditamap, AUTOSAR_MCAL<MDL>_IM.xml Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-20985	Bug	[MCL] Cannot create multiple DMA logical channels having the same DMA source<*> Detailed description (how to reproduce it): # Create a logical DMA channel in "MclConfigSet/MclConfigSet_0/DMAChannel" and set its "DmaSource0" to SRC_X*. # Create another logical DMA channel in "MclConfigSet/MclConfigSet_0/DMAChannel" and set its "DmaSource0" to the same SRC_X source. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Error: "Duplicate DMA Source". Expected behavior: No error given. Instead, maybe rise a warning. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Remove the check for duplicate DMA sources in xdm.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20998	Bug	[MCU] The range of FIRC Div2 Frequency should be 48k instead of 24k<*> Detailed description (how to reproduce it): Currently maximum Frequency of FIRC Div2 range in configuration is 24k. However in reference manual, Frequency of FIRC Div2 can be configured upto 48k. Preconditions: S32K14X 4.3 RTM 1.0.0 Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: FIRC Div2 Frequency can be configured up to 48k Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: fix limit node FIRC DIV 2 in file mcu.xdm
MCAL-21019	Bug	[PWM] u8ModuleId is wrongly used and causing an exception<*> Detailed description (how to reproduce it): in Pwm_Ftm_Init function, the u8ModuleId variable is used incorrect in this line of code: REG_BIT_CLEAR32(FTM_OUTMASK_ADDR32(u8ModuleId),(uint32)((uint32)1U<<u8HwChannel)); u8ModuleId shoud be replaced by u8HwModuleId as bellow: REG_BIT_CLEAR32(FTM_OUTMASK_ADDR32(u8HwModuleId),(uint32)((uint32)1U<<u8HwChannel)); Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21071	Bug	[ADC] ADC_ReadGroup work wrong if start 2 group without interrupt with the same number of channel and last channel<*> Detailed description (how to reproduce it): In S32k14x, the same HW unit we have two groups: Group 0: GroupID 0, AdcWithoutInterrupts enable, oneshot mode, priority 20, channel: AN0, AN1, AN2, AN3, AN4, AN5 Group 1: GroupID 1, AdcWithoutInterrupts enable, oneshot mode, priority 10, channel AN5, AN6 { ...... Adc_StartGroupConversion(0); Adc_StartGroupConversion(1); delay(); Adc_ReadGroup(1, DataPtr); ... }

Table continues on the next page...

ID	Subtype	Headline and Description
		Adc_ReadGroup returned E_OK when group 1 was not yet converted Preconditions: for S32k14X, number of channel of group0 gather than number of channel group1 for IPV_ADCDIG, tow groups had same last HW channel Test Case ID (internal TC that caught the defect) optional: [...] Observed behavior: group0 with higher priority convert first, group1 wait untill group0 stream complete, when call Adc_Readgroup for group1, this function only check data in result register of last HW channel group 1 (RB), but in RB contain data for group0 (it was convert yet) . So it return E_OK Expected behavior: Adc_ReadGroup return E_NOT_OK for group1 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: We must check group use by Adc_ReadGroup is first element in QUEUE or not before check data register
MCAL-21089	Bug	[MCU] Missing generated symbolic name for McuResetReason<*> Detailed description (how to reproduce it): As SWS item MCU186_Conf: McuResetReason has type is EcucIntegerParamDef and a symbolic name need to be generated for this parameter. However, currently MCU driver does not provide generated symbolic name for this parameter. Preconditions: S32K14X_MCAL4_0_RTM_1_0_3 Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: MCU driver does not provide generated symbolic name for McuResetReason parameter. Expected behavior: MCU driver provides generated symbolic name for McuResetReason parameter Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: The implement should be follow this requirement from AUTOSAR_TPS_ECUConfiguration: ecuc_sws_2108: The values of configuration parameters which are defined as symblicNameValue = true shall be generated into the header file of the declaring module as #define. The symbol shall be composed of - the module abbreviation <MA> of the declaring BSW Module (according to BswModuleList [17]) followed by the literal "Conf_" followed by - the shortName of the EcucParamConfContainerDef of the declaring modulefollowed by "_" followed by - the shortName of the EcucContainerValue container which holds the symblicNameValue configuration parameter value.
MCAL-21094	Bug	[OCU] The range of OcuChannelTickDuration should be from 1 to 32768 as CUC_Ocu_00153<*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Detailed description (how to reproduce it): As SWS item ECUC_Ocu_00153, the range of OcuChannelTickDuration should be from 1 to 32768. Currently in Ocu.xdm, the invalid range of OcuChannelTickDuration is from 0 to 65535. In epd file this range is also from 0 to 65535 and the default value is 0. So with AMDC checking, those error will be reported: Error;Ocu_s32k118_lqfp48.epd;Rule A207: Minimum value of parameter 'Ocu/OcuConfigSet/OcuChannel/OcuChannelTickDuration' in VSMD (0) may not be smaller than minimum value defined in StMD (1).; Error;Ocu_s32k118_lqfp48.epd;Rule A207: Maximum value of parameter 'Ocu/OcuConfigSet/OcuChannel/OcuChannelTickDuration' in VSMD (65535) may not be larger than maximum value defined in StMD (32768).; Preconditions: S32K14X_MCAL_4.2_RTM_1.0.1 Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Violate ASR requirement and error in AMDC checking. Expected behavior: no error in AMDC checking. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Update Ocu.xdm: Invalid range is from 1 to 32768. Default value should be in calculated in this range of OcuChannelTickDuration
MCAL-21102	Bug	[MCU] Mcu_RCM_GetResetReason does not return POR<*> Detailed description (how to reproduce it): As the note from Reference Manual from 24.4.3 System Reset Status Register (RCM_SRS): The reset value of this register depends on the reset source: - POR (including LVD) - 0x82 - LVD (without POR) - 0x02 - Other reset - a bit is set if its corresponding reset source caused the reset In Mcu_RCM_GetResetReason function, u32NumberOfFlags is 2 in case of POR (POR and LVD). So Mcu_RCM_GetResetReason return MCU_MULTIPLE_RESET_REASON instead of POR. Preconditions: S32K14x AUTOSAR MCAL 4.0 v1.0.3 POR occurs. Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Mcu_RCM_GetResetReason return MCU_MULTIPLE_RESET_REASON instead of POR. Expected behavior: Mcu_RCM_GetResetReason return POR Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-21104	Bug	[LIN] Frame identifiers 60 (0x3C) to 61 (0x3D) shall always use classic checksum<*> Detailed description (how to reproduce it): Lin spec for ASR 4.2 is base on LIN 2.1 specification. Follow 2.3.1.5 Checksum from LIN 2.1 specification: Enhanced in communication with LIN 2.x slave nodes Frame identifiers 60 (0x3C) to 61 (0x3D) shall always use classic checksum. 1. Currently LIN driver always uses enhanced checksum for all of frame: When Lin_SendFrame is called: enhanced checksum is fixed in Lin_LPUART_SendResponse function: au8LPUART_Buffer[u8Channel][au8LPUART_BufferLength[u8Channel] ((uint8)1U)] = (uint8)Lin_LPUART_ChecksumCalc(u8Channel, LIN_ENHANCED_CS); 2. In Lin_LPUART_GoToSleep, classic checksum is fixed. Preconditions: LIN SPEC 2.1 ASR 4.2 Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: The implementation does not follow the LIN 2.1 specification. Expected behavior: The LIN implementation should be follow the LIN 2.1 spec. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21105	Bug	[OCU] Add channel resource conflict checking<*> Detailed description (how to reproduce it): GPT,OCU,ICU,PWM can use a channel without ISR enable. Currently, FTM_Common.c and FTM_Common.c only check the conflict for ISR used. It is needed to replace ISR conflict check with Channel conflict check. Eg. [!FOR "ModuleIdx" = "0" TO "7"!] [!VAR "ChannelIsrUsed" = "'false'"!] [!LOOP "OcuGeneral/OcuHwResourceConfig/*"!] [!IF "OcuIsrEnable = 'true' and contains(OcuHwResourceId, concat('FTM_', $ModuleIdx))"!] [!VAR "ChannelIsrUsed" = "'true'"!] [!ENDIF!] [!ENDLOOP!] [!IF "$ChannelIsrUsed = 'true'"!] #define OCU_FTM_[!"$ModuleIdx"!]_ISR_USED [!ENDIF!] [!ENDFOR!] [!ENDIF!] Should be print file output #define OCU_FTM_0_USED #define OCU_FTM_1_USED The convention of channel used define is bellow : [!FOR "ModuleIdx" = "0" TO "7"!] [!VAR "ChannelIsrUsed" = "'false'"!]

Table continues on the next page...

ID	Subtype	Headline and Description
		[!LOOP "OcuGeneral/OcuHwResourceConfig/*"!] [!IF "OcuIsrEnable = 'true' and contains(OcuHwResourceId, concat('FTM_', $ModuleIdx))"!] [!VAR "ChannelIsrUsed" = "'true'"!] [!ENDIF!] [!ENDLOOP!] [!IF "$ChannelIsrUsed = 'true'"!] #define OCU_FTM_[!"$ModuleIdx"!]_USED [!ENDIF!] [!ENDFOR!] where [MDL] is OCU,PWM,ICU,GPT Similar to FTM... Preconditions: NA Test Case ID (internal TC that caught the defect) optional NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA
MCAL-21134	Bug	[MCU] McuSOSCDiv1Frequency and McuSOSCDiv2Frequency parameters have a wrong lower limit (4MHz)<*> Detailed description (how to reproduce it): I am trying to configure the System OSC (SOSC) to produce 750khz by configure "McuSOSCDiv2" . but i am always getting an error that this value is wrong and it must be modified to be above 4MHZ(as 4Mhz is the min. value to "McuSOSCDiv2Frequency") . I have checked the datasheet about this limitation. but, There was no limitation about this point . I found that there is a rule check that run on this configuration parameter and it has a limitation for McuSOSCDiv2Frequency that it should have value greater than or equal to 4MHz Note : i am using 24MHz crystal osc McuSOSCDiv1Frequency and McuSOSCDiv2Frequency parameters have a wrong lower limit (4MHz). Preconditions: Configure the System OSC (SOSC) to produce 750khz by configure "McuSOSCDiv2" Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Error in EB Expected behavior: No error Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Remove this parameter limit

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-21144	Bug	[CAN] Some registers are not cleared when perform init controller in ASR 4.2 and ASR4.0<*> Detailed description (how to reproduce it): Some registers do not be cleared when perform init controller in ASR 4.2 and ASR4.0. When Init controller, driver does not perform clear some registers, which are not affected by soft reset. Example : CAN_CTRL1, CAN_CTRL2, CAN_CBT. Preconditions: None Test Case ID (internal TC that caught the defect) optional: None Observed behavior: None Expected behavior: None Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Clear registers, which are not affected by soft reset before init controller.
MCAL-21145	Bug	[LINIF] Fix build error<*> Detailed description (how to reproduce it): LinIf_WakeupConfirmation, LinIf_CheckWakeup must use for AUTOSAR 4.2.1 and upper. Preconditions: NA Test Case ID (internal TC that caught the defect) optional NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): this is a debug ticket to create IPV_FLEXIO for LIN driver
MCAL-21176	Bug	[MCL] Reg_eSys_FlexIO.h available but not be documented in Integration Manual<*> Detailed description (how to reproduce it): Reg_eSys_FlexIO.h is available in Mcl_TS_T40D2M10I1R0\include but it is not documented in "3.2 Files required for Compilation" of Integration Manual. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Information in Integration Manual is not updated Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		Please review and update "3.2 Files required for Compilation" from IM. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Update 3.2 Files required for Compilation in IM
MCAL-21177	Bug	[LIN] Update information for integration manual<*> Detailed description (how to reproduce it): Chapter "3.2 Files required for Compilation" in Integration Manual should be updated. Example Reg_eSys_FlexIO.h file is not available in LIN module so it should not documented as: ..\ Lin_TS_T40D2M10I1R0\include\Reg_eSys_FlexIO.h Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: The information in Integration Manual is not updated Expected behavior: IM should be up to date Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: IM should be up to date
MCAL-21178	Bug	[I2C] Reg_eSys_FlexIO.h should be removed in I2C and use the file from MCL<*> Detailed description (how to reproduce it): Currently we have two file with name Reg_eSys_FlexIO.h used by I2C and LIN. Once from MCL and Once from I2C with the same content. This file should be removed from I2C and use the file from MCL. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: There are two files with the same name and same content in MCL and I2C Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Remove Reg_eSys_FlexIO.h from I2C and use the file from MCL. Update Integration Manual also for this chapter: "3.2 Files required for Compilation"
MCAL-21189	New	New Feature [LIN] Handle common interrupt for IPV_FLEXIO ,,"NewWorkDescription: Handle common interrupt for IPT_FlexIO Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Remove: Lin_FlexIO_Irq.c Modify: Lin.mak"
MCAL-21200	New	New Feature [I2C] Add support for "REPEATED START (Sr)"" ,,"NewWorkDescription: The client is requesting a "REPEATED START" option for I2C transfers in order to communicate with I2C slaves that require this sequence in their communication protocol. A START-after-STOP workaround will not reliably work, because the I2C Master will temporarily lose the bus between the STOP and the consequent START. Requirement source: Customer Request (*_Continental_*) (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: N/A

MCAL-21215	Bug	[CAN] Req CAN178 was not implemented correctly<*>
		
		Detailed description (how to reproduce it):
		The req CAN178 desciption that "The Can module may implement the function Can_MainFunction_Write as empty define in case no polling at all is used.". But in file Can.c, the function Can_MainFunction_Write was enclosed by condition like as "(CAN_TXPOLL_SUPPORTED == STD_ON)". For example :
		#if (CAN_TXPOLL_SUPPORTED == STD_ON)
		FUNC(void, CAN_CODE) Can_MainFunction_Write(void)
		{
		#if (CAN_DEV_ERROR_DETECT == STD_ON)
		/ Test whether the driver is already initialised. /
		if (CAN_UNINIT == Can_eDriverStatus)
		{
		/
		(CAN179) If development error detection for the module Can is enabled: The function Can_MainFunction_Write shall raise
		the error CAN_E_UNINIT if the driver is not yet initialized.
		/
		(void)Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_WRITE, (uint8)CAN_E_UNINIT);
		}
		else
		{
		#endif / (CAN_DEV_ERROR_DETECT == STD_ON) /
		/ Poll all controllers for Tx MB statuses. /
		Can_IPW_MainFunctionWrite();
		#if (CAN_DEV_ERROR_DETECT == STD_ON)
		}
		#endif / (CAN_DEV_ERROR_DETECT == STD_ON) /

Table continues on the next page...

ID	Subtype	Headline and Description

		}
		It mean in case "no polling", the function Can_MainFunction_Write is not defined_. It is not correct with req _CAN178
		The solution :
		We should enclose the condition for case "no polling" in this function. For example :
		FUNC(void, CAN_CODE) Can_MainFunction_Write(void)
		{
		#if (CAN_DEV_ERROR_DETECT == STD_ON)
		/ Test whether the driver is already initialised. /
		if (CAN_UNINIT == Can_eDriverStatus)
		{
		/
		(CAN179) If development error detection for the module Can is enabled: The function Can_MainFunction_Write shall raise
		the error CAN_E_UNINIT if the driver is not yet initialized.
		/
		(void)Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_WRITE, (uint8)CAN_E_UNINIT);
		}
		else
		{
		#endif / (CAN_DEV_ERROR_DETECT == STD_ON) /
		#if (CAN_TXPOLL_SUPPORTED == STD_ON)
		/ Poll all controllers for Tx MB statuses. /
		Can_IPW_MainFunctionWrite();
		#endif /*(CAN_TXPOLL_SUPPORTED == STD_ON) /
		#if (CAN_DEV_ERROR_DETECT == STD_ON)
		}
		#endif / (CAN_DEV_ERROR_DETECT == STD_ON) /
		}
		Preconditions:
		NA
		Test Case ID (internal TC that caught the defect) optional:
		[...]
		Observed behavior:
		NA
		Expected behavior:
		NA
		Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
		Proposed solution optional:
		[...]

MCAL-21234	New	New Feature [ADC] Optimize generate code for S32K14x ,,"In file Adc_RegOperations_42.m line 385 must be remove: [!VAR ""CRT_COUNT""=""$CRT_COUNT+1""!][!//"
MCAL-21247	Bug	[SPI] The default data is cast to uint8<*> Detailed description (how to reproduce it): The SpiDefaultData in configuration can be configure from 0 to 4294967295

Table continues on the next page...

ID	Subtype	Headline and Description
		(upto uint 32). However in the generated code, the default data is cast to uint8. So it causes confusing for customer when their configuration is not applied in the code. The SpiDefaultData should be compatible with the SpiDataWidth also. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: The SpiDefaultData is cast from uint32 to uint8. Expected behavior: Support the default data upto uint32 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: In file Spi.h ,Spi_RegOperations.m and Spi_RegOperations_42.m .DefaultTransmitValue value in struct Spi_ChannelConfigType is declared as uint32
MCAL-21250	Bug	[SPI] FLEXIO is reset all channel when call function Synctransmit<*> Detailed description (how to reproduce it): When driver use hw LPspi as master transmit and hw FLEXIO as slave receive . hw FLEXIO is reset when Synctranmit SQ use hw LPspi so now FLEXIO do not receive data ! Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: FLEXIO is reset all channel when call function Synctransmit Expected behavior: FLEXIO do not reset all channel when call function Synctransmit Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...]
MCAL-21273	Bug	[SPI] Incorrect transmit data when flexio channel in slave mode<*> Detailed description (how to reproduce it): Config FlexIO_1 channel is Slave mode, after that send data by call Spi_AsyncTransmit function. Issue: 1st byte is replaced by the 2rd byte Example: Data array are transmitted: uint8 MCP23S08_WRITE0_TEST[11]=\{0xAA,0xBB,0xCC,0xDD,0xEE,0xFF, 0x55,0x00,0x00,0x00,0x01}; Actual data on the bus (measured by Analyzer or oscilloscope): {0xBB,0xBB,0xCC,0xDD,0xEE,0xFF,0x55,0x00,0x00,0x00,0x01} Preconditions: Config FlexIO_1 is Slave mode Setting Spi_SetupEB Call Spi_AsyncTransmit

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) optional: Spi_TC_0151 (SPI_TS_023) Observed behavior: 1st byte is replaced by the 2rd byte Expected behavior: Data is transmitted correctly Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...]
MCAL-21274	Bug	[ETH] Incorrect value of ENET_ECR_DEFAULT_RESERVED_VALUE_U32<*> Detailed description (how to reproduce it): in Eth_Enet.h, ENET_ECR_DEFAULT_RESERVED_VALUE_U32 is defined as 0x70000000U. However, the default value should be 0xF0000000U as reference manual. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Incorrect the value of ENET_ECR_DEFAULT_RESERVED_VALUE_U32 Expected behavior: Update the value of ENET_ECR_DEFAULT_RESERVED_VALUE_U32 to 0xF0000000 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21277	Bug	[MCU] SLEEPDEEP bit is not clear after waked-up from VLPS or STOP mode<*> Detailed description (how to reproduce it): Currently Mcu_SetMode function calls Call_Mcu_CM4_EnableDeepSleep to set SLEEPDEEP bit to 1. However when MCU is wake-up, this bit still remain the value is 1. So it could cause unexpected sleep. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: DEEPSLEEP bit still remain set state after wakeup from VLPS or STOP mode. Expected behavior: DEEPSLEEP bit should be clear after exit from VLPS/STOP mode Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: add function in Mcu_CortexM4.c FUNC(void, MCU_CODE) Mcu_CM4_DisbleDeepSleep(void) { / violates ref Mcu_CortexM4_c_REF_4 Required Rule 11.1, Conversion from integer to pointer /

Table continues on the next page...

ID	Subtype	Headline and Description

		/ violates ref Mcu_CortexM4_c_REF_5 The cast is used to access memory mapped registers.*/
		REG_RMW32(CM4_SCR_BASEADDR, CM4_SCR_SLEEPDEEP_MASK32, 0x00U);
		}
		
		and call this function in Mcu_IPW.c bellow
		Call_Mcu_SMC_ModeConfig(Mcu_pModeConfigPtr);
		 if ((MCU_STOP1_MODE == Mcu_pModeConfigPtr->u32PowerMode)
		(MCU_STOP2_MODE == Mcu_pModeConfigPtr->u32PowerMode)
		(MCU_VLPS_MODE == Mcu_pModeConfigPtr->u32PowerMode))
		{
		Call_Mcu_CM4_DisbleDeepSleep();
		}

MCAL-21287	Bug	[GPT] Missing the ISR information for S32K11x<*> Detailed description (how to reproduce it): Missing the ISR information of S32K11x in chapter 5.2: "ISR to configure within OS - dependencies" from Integration Manual. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Update information ISR to configure within OS dependencies chapter from UM. Expected behavior: the ISR information of S32K11x in chapter 5.2: "ISR to configure within OS - dependencies" available on intergration Manual. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21290	Bug	[SPI] Data is not received when number data is less than FCR<*> Detailed description (how to reproduce it): In Spi_RegOperations.m ,the value for FCR always is generated with this value: (uint32)(((uint32)(3) << 16) ((uint32)(0))). It mean that FCR value always is 3. So number data received is greater than FCR value then receive data flag is ready Preconditions: S32K14X_MCAL4_0_RTM_1_0_3 Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Data is not received when number data is less than FCR Expected behavior: Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Edit Spi_RegOperations.m file ,the value for FCR always is generated with this value:

Table continues on the next page...

ID	Subtype	Headline and Description
		(uint32)(((uint32)(0) << 16) ((uint32)(0))).
MCAL-21294	Bug	[LIN] Master need support both type checksum<*> Detailed description (how to reproduce it): As Autosar Lin Driver only support Master and Lin spec 2.x and in the LIN 2.x document describe "classic in communication with LIN 1.3 slave nodes and enhanced in communication with LIN 2.0 slave nodes." ==> LIN Driver need support both type classic and enhanced checksum. But in Lin_LPUART, Lin_SCI, Lin_FlexIO ipv only support enhanced checksum type in the sendframe. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: need update Driver support both type checksum Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: refer to Lin_LinFlex for sample
MCAL-21302	Bug	[LIN] Issue when send data length more than 8 bytes<*> Detailed description (how to reproduce it): When send a frame with data length more than 8 bytes, driver may be hard fault. Issue: Driver still write data into data array: au8LPUART_Buffer[LIN_HW_MAX_MODULES] [LPUART_MAX_BUFFER_LENGTH_U8]; => Write over memory Preconditions: Send a frame with data length more than 8 bytes Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Write over memory Expected behavior: Check range of data (0-8 bytes) Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Verify data length when push into buffer
MCAL-21310	Bug	[LIN] Behavior on driver is wrong in Slave to Slave case<*> Detailed description (how to reproduce it): According to AUTOSAR spec version 4.2.2 (and same for others), we have some points as below: 1. The LIN driver applies to LIN 2.1 master nodes only. Operating as a slave node is out

Table continues on the next page...

ID	Subtype	Headline and Description
		of scope. (Page 6 1.1 Scope) 2. [SWS_Lin_00230] LIN_SLAVE_TO_SLAVE Response is generated from one slave to another slave, for the master the response will be anonymous, it does not have to receive the response. (Page 35 8.2.4 Lin_FrameResponseType) 3. [SWS_Lin_00238] [The function Lin_GetStatus shall return LIN_TX_OK, when A Master Response Type frame is send and LIN header as well as LIN response of the frame are transmitted successfully or A Slave to Slave Response Type frame is send and the LIN header of the frame is transmitted successfully.] (Page 46 8.3.2.6 Lin_GetStatus) However, for Slave to Slave case (pPduInfoPtr->Drc == LIN_SLAVE_TO_SLAVE), driver handled: 1. Still receive the response and copy data to Lin_SduPtr[] 2. Lin_GetStatus() function return LIN_RX_OK instead LIN_TX_OK Preconditions: Call Lin_Sendframe with PDU: pPduInfoPtr->Drc = LIN_SLAVE_TO_SLAVE Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Wrong behavior Expected behavior: Follow to AUTOSAR spec Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: TBC
MCAL-21313	Bug	[ETH] u32CorrectCycle parameter is casted to uint8<*> Detailed description (how to reproduce it): u32CorrectCycle parameter is casted to uint8 in Eth_Enet_SetCorrectionTime function. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: the s32CorrectCycle argument passing inside Eth_Enet_CorrectRate() is being typecasted to uint8,as ATCOR register is 32 bit so it is truncating the value of s32CorrectCycle hence it is not performing clock correction accurately. Expected behavior: s32CorrectCycle should not be casted to uint8 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21317	Bug	[FLS] Wrong page write description in IM<*> emphasized text__*Detailed description (how to reproduce it): When a write is cancelled right after it starts, it may write some values, depending on the delay. In the manual it is stated that those values are written from higher to lower address, but the tests show the opposite.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: Start a write and cancel it almost immediately, then read the written value. Test Case ID (internal TC that caught the defect) optional: [...] Observed behavior: The value is written from lower to higher address, opposed to the information in the manual. Expected behavior: The manual should describe the exact behavior of the hardware. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Correct manual entry.
MCAL-21319	Bug	[SPI] Ipw_JobsCount is 0 when use Spi_Ipw_SyncTransmit() function with Hw LPspi<*> Detailed description (how to reproduce it): BLN_SMCAL_4.2_S32K14X_RTM_P3_1.0.1 is defined to support both of FLEXIO and LPSPI. However, at Spi_Ipw_SyncTransmit(), Ipw_JobsCount only init if Flexio defined/enabled. Here we are block of code: #if(defined FLEXIO_0_SPI_0_USED defined FLEXIO_0_SPI_1_USED) #if ((FLEXIO_0_SPI_0_USED == STD_ON) (FLEXIO_0_SPI_1_USED == STD_ON)) VAR(uint32, AUTOMATIC)SPI_REG_FLEXIO; / Get sequence configuration / / violates ref Spi_IPW_c_REF_6 Array indexing shall be the only allowed form of pointer arithmetic. */ Ipw_pcSequenceConfig = &Spi_pcSpiConfigPtr->pcSequenceConfig[Sequence]; / Get the number of jobs in the sequence / Ipw_JobsCount = Ipw_pcSequenceConfig->NumJobs; Ipw_pcJobIndexList = Ipw_pcSequenceConfig->pcJobIndexList; #endif #endif Preconditions: BLN_SMCAL_4.2_S32K14X_RTM_P3_1.0.1 LPSPI syntransmit Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Ipw_JobsCount is 0 when use Spi_Ipw_SyncTransmit() function with Hw LPspi Expected behavior: Ipw_JobsCount init with hw LPspi and FlexIO Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: In file Spi_IPW.c line 214 edit code #if(defined FLEXIO_0_SPI_0_USED defined FLEXIO_0_SPI_1_USED) #if ((FLEXIO_0_SPI_0_USED == STD_ON) (FLEXIO_0_SPI_1_USED == STD_ON)) VAR(uint32, AUTOMATIC)SPI_REG_FLEXIO; #endif

Table continues on the next page...

ID	Subtype	Headline and Description
		#endif / Get sequence configuration / / violates ref Spi_IPW_c_REF_6 Array indexing shall be the only allowed form of pointer arithmetic. */ Ipw_pcSequenceConfig = &Spi_pcSpiConfigPtr->pcSequenceConfig[Sequence]; / Get the number of jobs in the sequence / Ipw_JobsCount = Ipw_pcSequenceConfig->NumJobs; Ipw_pcJobIndexList = Ipw_pcSequenceConfig->pcJobIndexList;
MCAL-21324	New	New Feature [ADC] AdcDelayNextPdb should be noticed to configure for the group with more than 8 channels ,,"NewWorkDescription: In case of group is configured with more than 8 channels, if AdcDelayNextPdb is 0 then this group will does not work. However, there is nothing reminds customer to notice this. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: N/A"
MCAL-21333	New	New Feature [SPI] Support pin configuration PINCFG from CFGR1 register ,,"NewWorkDescription: As RM: Pin Configuration Configures which pins are used for input and output data during 1-bit transfers. If performing 2-bit or 4-bit transfers, the Pin Configuration field must be 00. 00b SIN is used for input data and SOUT is used for output data 01b SIN is used for both input and output data 10b SOUT is used for both input and output data 11b SOUT is used for input data and SIN is used for output data Currently driver does not support to configure PINCFG from CFGR1 register. SPI use the default mode: SIN is used for input data and SOUT is used for output data. This support might useful with customer. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: N/A"
MCAL-21447	Bug	[MCEM] Missing code for FILE VERSION CHECKS in Mcem.h<*> Detailed description (how to reproduce it): Missing code to check version of file in Mcem.h For example :

Table continues on the next page...

ID	Subtype	Headline and Description
		/ *=== ===================================== SOURCE FILE VERSION INFORMATION === =====================================*/ / *=== ===================================== FILE VERSION CHECKS === =====================================*/ Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Missing code to check version of file in Mcem.h Expected behavior: Having code to check version of file for Mcem.h Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: before : / *=== ===================================== SOURCE FILE VERSION INFORMATION === =====================================*/ / *=== ===================================== FILE VERSION CHECKS === =====================================*/ After{color}: / *=== ===================================== SOURCE FILE VERSION INFORMATION === =====================================*/ #define MCEM_VENDOR_ID M4_SRC_AR_MODULE_VENDOR_ID #define MCEM_MODULE_ID M4_SRC_AR_MODULE_ID`U' / violates ref CDD_Mcem_h_REF_2 Exceeds 31 characters / #define MCEM_AR_RELEASE_MAJOR_VERSION M4_SRC_AR_SPEC_VERSION_MAJOR / violates ref CDD_Mcem_h_REF_2 Exceeds 31 characters / #define MCEM_AR_RELEASE_MINOR_VERSION M4_SRC_AR_SPEC_VERSION_MINOR / violates ref CDD_Mcem_h_REF_2 Exceeds 31 characters / #define MCEM_AR_RELEASE_REVISION_VERSION M4_SRC_AR_SPEC_VERSION_PATCH #define MCEM_SW_MAJOR_VERSION M4_SRC_SW_VERSION_MAJOR #define MCEM_SW_MINOR_VERSION M4_SRC_SW_VERSION_MINOR #define MCEM_SW_PATCH_VERSION M4_SRC_SW_VERSION_PATCH

Table continues on the next page...

ID	Subtype	Headline and Description
		/ *=== ====================================== FILE VERSION CHECKS === ======================================*/ / Check for CDD_Mcem.h versions / / Check if current file and CDD MCEM header file are of the same vendor / #if (MCEM_VENDOR_ID != CDD_MCEM_VENDOR_ID) #error "Mcem.h and CDD_Mcem.h have different vendor ids" #endif / Check if current file and MCEM configuration header file are of the same Autosar version / #if ((MCEM_AR_RELEASE_MAJOR_VERSION != CDD_MCEM_AR_RELEASE_MAJOR_VERSION) \\ (MCEM_AR_RELEASE_MINOR_VERSION != CDD_MCEM_AR_RELEASE_MINOR_VERSION) \\ (MCEM_AR_RELEASE_REVISION_VERSION != CDD_MCEM_AR_RELEASE_REVISION_VERSION)) #error "AutoSar Version Numbers of Mcem.h and CDD_Mcem.h are different" #endif / Check if current file and MCEM configuration header file are of the same software version / #if ((MCEM_SW_MAJOR_VERSION != CDD_MCEM_SW_MAJOR_VERSION) \\ (MCEM_SW_MINOR_VERSION != CDD_MCEM_SW_MINOR_VERSION) \\ (MCEM_SW_PATCH_VERSION != CDD_MCEM_SW_PATCH_VERSION)) #error "Software Version Numbers of Mcem.h and CDD_Mcem.h are different" #endif
MCAL-21450	New	New Feature [SPI] Integrative FLEXIO SPI from S32K2 and Improve some wrong codes for S32K1 release ,,"NewWorkDescription: Integrative FLEXIO SPI from S32K2 and Improve some wrong codes for S32K1 release Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: need to improve. need to refer this pullrequest for more information: https://bitbucket.sw.nxp.com/projects/AMNG/repos/spi/pull-requests/87/overview More detail: In Reg_eSys_FlexIOSPI.h: Every defines need to add ""SPI_"" at the beginning of each define name In Spi.xmd This delay is important for SPI protocol, please add a note into description to mention about this delay. How to know the value of this delay? SpiPhyTxDmaChannelAux is only used if DSPI is used. So we can add a note into description of that node. In Spi_RegOperations.m [!MACRO ""Spi_AttributeConfigurationOfChannels""!] need to improve eg: [!IF ""$Exist_LPSPI = 1""!]

Table continues on the next page...

ID	Subtype	Headline and Description
		[!CALL ""Spi_AttributeConfigurationOfChannels_LPSPI""!] [!ELSE!] [!CALL ""Spi_AttributeConfigurationOfChannels_LPSPI_NULL""!] [!ENDIF!] In Spi_FlexIO.c Reset value should be defined via a macro and used to write into the register. Should not use hard code. eg: REG_WRITE32(FLEXIO_TIMCTLn_ADDR32((uint32) Spi_FlexIO_u8Timer_CLK), FLEXIO_TIMCTL_RESET_U32); No need to check master slave with SyscTransmit mode, because Slave is only supported in AsyncTransmit mode. This should be compared with the define SPI_FLEXIO_<LSB/MSB>_U8 This comment is also applied for Spi_FlexIO_SyncTransmitRX function. In the case of transfer MSB first, this will not work when the data width is 16 or 32 bits. This comment is also applied for Spi_FlexIO_SyncTransmitRX function. Need to add else case for SPI_FLEXIO_CPOL_LOW_U8. Should use define name SPI_FLEXIO_CPOL_HIGH_U8 and SPI_FLEXIO_CPHA_TRAILING_U8 in check conditions above instead of hard code. Rule 3.15 The constants shall be placed on the left of equality comparisons. All defines with the prefix SPI_FLEXIO_.... like SPI_FLEXIO_DATA_WIDTH_8, SPI_FLEXIO_LSB_U8, SPI_FLEXIO_CPOL_LOW_U8 etc should be moved to IP layer I think this can be improved to use the condition ((Spi_FlexIO_pSyncTransmitState->Spi_FlexIO_LengthTX>0u) (Spi_FlexIO_pSyncTransmitState->Spi_FlexIO_LengthRX>0u)) CTRL register cannot be used to clear Shifter. This will clear all registers, so it will affect other modules that are using FlexIO. Please remove it. Spi_FlexIO_ChannelTransferDmaInitTX: Don't use global variable Spi_FlexIO_AsyncTransmitState. Because they will be modified in interrupt function. Spi_FlexIO_IsrFifoRx: Spurious interrupt needs to be handled when the driver wasn't initialized and enable interrupt bits weren't set. Spi_FlexIO_TransferStatus: In the case of RX: No needs to initialize for next Channel with Slave mode. Beacause Slave only support one Channel, one Job, one Sequence. Why do the driver need to re-enable interrupt for TX shifter? I think it is redundant and don't enable it in POLLING_MODE. In the case of TX, Why do the driver need to disable interrupt for TX shifter? Spi_FlexIO_JobTransferFifoDrain: Spi_FlexIO_nFramesCount is always equal to 1 so: No needs to check for Slave mode at the beginning of the function. No needs to use do-while statement. The driver doesn't support to receive MSB first. Spi_FlexIO_JobTransferFifoFill: Spi_FlexIO_nFramesCount is always equal to 1 so no needs to use do-while statement. The driver doesn't support to transmit MSB first. In Spi_FlexIO.h Struct fields are still using the name of LPSPI's registers name. In Spi_Cfg.h We should add 2 defines like SPI_<DSPI/LPSPI>_USED. And these defines will be used to guard for DSPI/LPSPI/FLEXIO's functions. So the code size after compile will be reduce if one of them is not used.

Table continues on the next page...

ID	Subtype	Headline and Description
		Update some comments in the code line to get more their means > Please see MCAL-21716 MCAL-21715 MCAL-21717 for the feature will be improved"
MCAL-21455	Bug	[CAN] Wrong path for attribute INVALID in reference CanCpuClockRef_Alternate<*> Detailed description (how to reproduce it): Reference CanCpuClockRef_Alternate have a wrong path in INVALID attribute: <a:da name="INVALID" type="XPath" expr="(../../../../../CanGeneral/CanEnableDualClockMode = 'true') and (not(node:refvalid(.)))" true="The configured node does not exist or may not be referenced."/> Preconditions: enable CanEnableDualClockMode and select CanCpuClockRef_Alternate to a invalid ref but driver did not raise error on EB tresos. Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Correct path which refer to CanEnableDualClockMode in INVALID attribute
MCAL-21468	Bug	[FLS] Flash data size calculated incorrectly in Fls_Flash_GetFProtReg function<*> Detailed description (how to reproduce it): checking patition does not be enough for S32K144 For ex: when FlexNVM partition data flash size with 32kb data and 32kb eeprom(DEPART = 1011). then call Fls_Flash_GetFProtReg function please see attachment for more detail Preconditions: When run test on S32K144 with DEPART = 1011 Test Case ID (internal TC that caught the defect) optional: Fls_TC_00202.c Observed behavior: test failed Expected behavior: checking all partition for all DERIVATIVE Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: should check each separate derivative to make sure correctly.
MCAL-21601	Bug	[CAN] CAN_TXPOLL_SUPPORTED/CAN_TXPOLL_SUPPORTED were defined not correctly when checking spurious interrupt<*> Detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description

		CAN_TXPOLL_SUPPORTED/CAN_TXPOLL_SUPPORTED were defined not correctly when checking spurious interrupt with more than one controller in config_set.
		#define CAN_RXPOLL_SUPPORTED [!WS "1"!][!IF "$rx_pol_en = 1"!](STD_ON)[!ELSE!](STD_OFF)[!ENDIF!]
		$rx_pol_en = 1 if at least 1 controller setting as POLLING.
		So if we have 2 controllers in 1 config_set, 1 controller as POLLING, 1 controller as INTERRUPT. > $rx_pol_en = 1 > CAN_RXPOLL_SUPPORTED = STD_ON.
		Therefore, these statements below will be never executed in this case:
		#if (CAN_TXPOLL_SUPPORTED == STD_OFF)
		/*Get the content of IMASK register for which corresponds this MB. /
		u32ImaskReg = REG_READ32(Can_IflagImask[u8IflagRegIndex][u8HwOffset].u32CanImask);
		/ Check for spurious interrupt. /
		if (((uint32)CAN_CONTROLLERCONFIG_TXPOL_EN_U32 == (CAN_CONTROLLERCONFIG_TXPOL_EN_U32 & CanStatic_pControlerDescriptors[controller].u32Options))
		((uint32)u32TempFlag == (u32ImaskReg & u32TempFlag)))
		{
		#endif
		
		#if (CAN_RXPOLL_SUPPORTED == STD_OFF)
		/*Get the content of IMASK register for which corresponds this MB. /
		u32ImaskReg = REG_READ32(Can_IflagImask[u8IflagRegIndex][u8HwOffset].u32CanImask);
		/ Check for spurious interrupt in Interrupt mode /
		if (((uint32)CAN_CONTROLLERCONFIG_RXPOL_EN_U32 == (CAN_CONTROLLERCONFIG_RXPOL_EN_U32 & CanStatic_pControlerDescriptors[controller].u32Options))
		((uint32)u32TempFlag == (u32ImaskReg & u32TempFlag)))
		{
		#endif
		
		Preconditions:
		None
		Test Case ID (internal TC that caught the defect) optional:
		None
		Observed behavior:
		None
		Expected behavior:
		None
		Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
		Proposed solution optional:
		None

MCAL-21604	Bug	[I2C] Missing requirement for I2c in slave mode support repeated start (Sr) condition<*> NewWorkDescription: Missing requirement for I2c in slave mode support repeated start (Sr) condition Requirement source: None (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional:

Table continues on the next page...

ID	Subtype	Headline and Description
		None
MCAL-21610	New	New Feature [ADC] Continuous conversion without interrupt doesn't work properly ,,"NewWorkDescription: In Adc_TS_005_cfg1 development test Adc_group0 is configured in SW continuous without interrupt DMA mode. This group only converts once while it's expected to continously convert Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Move function Adc_Pdb_StopConversion() of Adc_Adc12bsarv2_StartNormalConversion() to a suitable place to avoid reset continous bit CONT in PDB SC register (PDB_SC_CONT) after it was properly set"

MCAL-21613	Bug	[SPI] Rx buffer cannot receive data in Master Mode<*>
		
		Detailed description (how to reproduce it):
		When Maf(SLAVE) transmit data for Mother_Board(Master) then Master cannot recive data that Slave send.
		Don't use correct function in Mcl plugin.
		Preconditions:
		In Master mode.
		Test Case ID (internal TC that caught the defect) optional:
		tc_wir_spi_00100.c
		Observed behavior:
		NA
		Expected behavior:
		Side Master recive data which send to by Maf.
		Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
		Proposed solution optional:
		Update in Spi_FlexIO_SyncTransmitJob function in Spi_FlexIO.c file as below:
		if(0u != Mcl_Flexio_ReadShiftStat(1<<Spi_FlexIO_u8Shifter_TX))
		{
		SPI_FlexIO_TransStatus =
		Spi_FlexIO_SyncTransmitTX(Spi_FlexIO_pSyncTransmitState);
		if((Std_ReturnType)E_OK == SPI_FlexIO_TransStatus)
		{
		/ reset the timeout /
		Spi_FlexIO_u32Timeout = SPI_TIMEOUT_COUNTER;
		}
		else
		{
		/ Do nothing /
		}
		}
		/ Receive /
		/
		violates ref Spi_FlexIO_c_REF_3 A cast should not be performed
		between a pointer type and an integral type.

Table continues on the next page...

ID	Subtype	Headline and Description

		/
		/
		violates ref Spi_FlexIO_c_REF_12 A cast should not be performed between a pointer type and an integral type.
		/
		/
		violates ref Spi_FlexIO_c_REF_5 The value of a composite expression shall not be cast to a different
		essential type category or a wider essential type Pointer type conversions.
		/
		if(0u != Mcl_Flexio_ReadShiftStat(1<<Spi_FlexIO_u8Shifter_RX))
		{
		SPI_FlexIO_TransStatus = Spi_FlexIO_SyncTransmitRX(Spi_FlexIO_pSyncTransmitState,SPI_FLEXIO_SYNCTRANSMIT);
		if((Std_ReturnType)E_OK == SPI_FlexIO_TransStatus)
		{
		/ reset the timeout /
		Spi_FlexIO_u32Timeout = SPI_TIMEOUT_COUNTER;
		}
		else
		{
		/ Do nothing /
		}
		}

MCAL-21620	Bug	[SPI] Has wrong code for DMA mode in LPSPI IP vault<*> NewWorkDescription: The wrong code for DMA mode in LPSPI Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: In Spi_LPspi_ChannelTransferDmaInitTX and Spi_LPspi_ChannelTransferDmaInitRX: Has duplicated this line code : if (SPI_DATA_WIDTH_32 == LPspi_pcChannelAttributesConfig->u8DataWidth)
MCAL-21644	New	New Feature [ADC] Lack of checking conversion complete flags in ISR function ,,"NewWorkDescription: Adc_Adc12bsarv2_EndGroupConversion() only checks interrupt mask (AIEN bit) but not check its respective interrupt flag (COCO bit). This behaviour doesn't meet requirement: [PR-MCAL-3272] 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 of the flag is not set, then the interrupt is spurious and the ISR shall only clear interrupt status flag and return immediately. Moreover, incase of spurious interrupt occurs before ADC conversion completes, ISR function will still improperly record the result and effect FMEA.

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Add checking condition for both AIEN and COCO at the same time."
MCAL-21646	New	New Feature [ADC] Duplicated channels in AdcGroupDefinition are no longer be applied ,,"NewWorkDescription: Driver will not support to configure channels with same channel ID in a group. Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: NA"

MCAL-21668	New	New Feature
		
		[ADC] Improperly get SC1 register values in ISR function
		,,"NewWorkDescription:
		There is a FOR loop in Adc_Adc12bsarv2_EndGroupConversion:
		for(u8ResultReg = 0U; u8ResultReg < Adc_aUnitStatus[Unit].u8Sc1Used; u8ResultReg++)
		{
		/ Read SC1n registers /
		u32SC1Value = REG_READ32(ADC12BSARV2_SC1_REG_ADDR32(Unit, (uint32)u8ResultReg));
		/ Check interrupt enable bit in SC1 /
		if(ADC12BSARV2_CONV_COMPLETE_INT_EN_U32 == (u32SC1Value & ADC12BSARV2_CONV_COMPLETE_INT_EN_U32))
		{
		break;
		}
		}
		For each loop, current data of SC1 value is improperly read because it's ""OR""ed with the previous u32SC1Value results.
		It will make conversion complete flag (COCO) is not correct incase of spurious interrupt occurs before the last channel is completely converted.
		Because COCO flag data is belong to one (or some) flags of the previous channels but not the last channel.
		Requirement source:
		NA
		(e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
		Proposed solution optional:
		Replace ""|="" to only ""=""

MCAL-21673	Bug	[ICU] Timestamps not working correctly in ICU withDMA<*> Detailed description (how to reproduce it): The timestamp is working correctly with the captured edges (time correctly obtained), but it does NOT restart to the destination address buffer when it reaches its limit (size already set in API Icu_StartTimestamp).

Table continues on the next page...

ID	Subtype	Headline and Description
		The DMA return index is correctly working as circular buffer (checked with Icu_GetTimestampIndex API) but the DMA is not resetting the destination address in register TCD1_DADDR after the DMA major loop count is completed. The DMA keeps increasing the destination address forever until it fills all the RAM memory and a hard fault is raised in the microcontroller. Steps performed: Initialization: 1. Icu_Init(&IcuConfigSet) 2. Icu_StartTimestamp(IcuChannel_0, (uint32)icu_buffer, 128, 0); Periodic: 3. icu_result_index = Icu_GetTimestampIndex(IcuChannel_0); Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Hard fault Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21677	Bug	[CAN] HardFault_Handler exception when reset controller not supported FD<*> Detailed description (how to reproduce it): HardFault_Handler exception occurs when reset controller not supported FD on S32K146/S32K144/S32K142 platform. REG_WRITE32(FLEXCAN_FDCBT(u8HwOffset), FLEXCAN_FDCBT_DEFAULT_VALUE_U32); for asr4.0, 4.2 In fact, no all controllers support FD mode, but in Can_FlexCan_ResetController function, driver always clear FDCTRL, FDCBT registers which are not supported. for asr4.3, in Can_DeInit driver also always clears FDCTRL, FDCBT Registers for all controllers, even when, not all of them support FD. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: [...] Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...]
MCAL-21717	New	New Feature [SPI] Improve FLEXIO IP vault ,,"NewWorkDescription: Improve FLEXIO IP vault

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Spi.xdm: 1. This delay is important for SPI protocol, please add a note into description to mention about this delay. How to know the value of this delay? Need to add the information for FlexIO (Baudrate) Spi_IPW.h: 2. I don't find out where these defines is using. what is the purpose of these defines? Reg_eSys_FlexIOSPI.h: 3. Every defines need to add ""SPI_"" at the beginning of each define name You should check with all defines in Reg_eSys_FlexIO.h of MCL. Currently I see that so many defines are duplicate. Spi_RegOperations.m: 4. this can be improved as below: > Have the problem with multiconfig [!IF ""$Exist_LPSPI = 1""!] [!CALL ""Spi_AttributeConfigurationOfChannels_LPSPI""!] [!ELSE!] [!CALL ""Spi_AttributeConfigurationOfChannels_LPSPI_NULL""!] [!ENDIF!] .....same for DSPI and FlexIO 5. DualClock Mode is not supported for FlexIO > Has a ticket to develop this Spi_FlexIO.h: 6. We should have a ticket to improve all comments for IPV_FLEXIO Spi_Cfg.h: 7. FLEXIOSPI_[!""num:i($Number)""!]_OFFSET value may equal with LPSPI_x_OFFSET, so each bit in Spi_au32SpiSeqUsedHWUnits[x] and Spi_u32SpiBusySyncHWUnitsStatus will map to wrong SPI HW unit, so SyncTransmit function will not work properly. Spi_FlexIO.c: 8. No need to use 2 variable Spi_FlexIO_AsyncTransmitState, Spi_FlexIO_SyncTransmitState, you can use one variable of them. Can improve to merge into Spi_FlexIO_aDeviceState variable. 9. An empty line must be inserted after the variable declaration and before the code 10. reset value should be defined via a macro and used to write into the register. Should not use hard code. eg: REG_WRITE32(FLEXIO_TIMCTLn_ADDR32((uint32) Spi_FlexIO_u8Timer_CLK), FLEXIO_TIMCTL_RESET_U32); 11. FlexIO_SPI driver did not support SyncTransmit_Fast. > Has a ticket to develop this 12. All FLEXIO's functions and variables should be guarded by the SPI_FLEXIO_USED define. So the code size after compile will be reduce if one of them is not used. 13. Spi_FlexIO_SlaveCancel: should not use FLEXIO_CTRL_ADDR32 to make software reset 14. This should be compared with the define SPI_FLEXIO_<LSB/MSB>_U8. 15. In the case of transfer MSB first, this will not work when the data width is 16 or 32 bits. 16. Should use define name SPI_FLEXIO_CPOL_HIGH_U8 and SPI_FLEXIO_CPHA_TRAILING_U8 in check conditions above instead of hard code. 17. Rule 3.15 The constants shall be placed on the left of equality comparisons. 18. this can be improved to use the condition

Table continues on the next page...

ID	Subtype	Headline and Description
		((Spi_FlexIO_pSyncTransmitState->Spi_FlexIO_LengthTX>0u) (Spi_FlexIO_pSyncTransmitState->Spi_FlexIO_LengthRX>0u)) 19. Spurious interrupt needs to be handled when enable interrupt bits weren't set. 20. Spi_FlexIO_TransferStatus: Why do the driver need to re-enable interrupt for TX shifter? 21. Spi_FlexIO_JobTransferFifoDrain: The driver doesn't support to receive MSB first. 22. Spi_FlexIO_JobTransferFifoFill: The driver doesn't support to transmit MSB first."
MCAL-21722	New	New Feature [MCU] Implement changes according to new RM for S32K1xx ,,"NewWorkDescription: Implement changes according to new RM for S32K14x RTM 1.0.4 ASR 4.0 Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Update values and description of SCG_FIRCCFG[RANGE]."
MCAL-21734	Bug	[MCL] Define for field in LPTMR_CSR is wrong<*> Detailed description (how to reproduce it): Define LPTMR_CSR_TPS_MASK_U32 is wrong in the common Lptmr files from MCL. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: The timer pin select should correctly write the value in the field LPTMR_CSR[TPS] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Update define LPTMR_CSR_TPS_MASK_U32 from file Reg_eSys_Lptmr.h from ((uint32)BIT5|BIT14) to ((uint32)BIT5|BIT4)
MCAL-21737	New	New Feature [ADC] AdcGroupPriority is still configurable in ADC_PRIORITY_NONE mechanism ,,"NewWorkDescription: AdcGroupPriority is still configurable when AdcPriorityImplementation node is ADC_PRIORITY_NONE Requirement source: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		(e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Correct the XPATH in AdcGroupPriority <a:a name=""EDITABLE"" type=""XPath""> <a:tst expr=""../../../../../../AdcGeneral/AdcPriorityImplementation != 'ADC_PRIORITY_NONE'""/> </a:a>"
MCAL-21738	Bug	[ADC] AdcConvTimeOnce and AdcEnableDualClockMode works incorrectly<*> Detailed description (how to reproduce it): When not in use AdcConvTimeOnce then users can still configure AdcNormalConversionTimings in hardware unit and AdcGroupAlternateConversionTimings in group. When not in use AdcConvTimeOnce and use AdcEnableDualClockMode then users can still configure AdcNormalConversionTimings and AdcAlternateConversionTimings in hardware unit. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: Adc_TC_0206 Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21739	New	New Feature [ICU] Support nonAutosar STANDBY operation as requested on CPR-MCAL-880 ,,"NewWorkDescription: Add support for wakeup accross STANBY as requested by CPR-MCAL\-880 and implemented on HF branch of tickets : MCAL-20077|[HF][ICU] Add support for wakeup source reporting after STANDBY wakeup MCAL-19299|[HF] [ICU] Add support for wakeup source reporting after STANDBY wakeup Requirement source: CPR-MCAL\-879 (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [NA]"
MCAL-21746	Bug	[ICU] Remove defines like ICU_LPTMR_CSR_TPS_MASK_U32<*> Detailed description (how to reproduce it): Remove defines like ICU_LPTMR_CSR_TPS_MASK_U32 from the Icu_Lptmr_Types.h and use the coresponding defines from MCL driver (they are duplicated in ICU driver)

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: Test Case ID (internal TC that caught the defect) optional: tc_fnc_icu_002110 tc_fnc_icu_002111 Observed behavior: Expected behavior: Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Remove defines like ICU_LPTMR_CSR_TPS_MASK_U32 from the Icu_Lptmr_Types.h and use the coresponding defines from MCL driver
MCAL-21747	New	New Feature [MCU] Support configuration parameters to disable parts of MCU Initialization ,,"MCU: Support configuration parameters McuDisableClockInit McuDisableResetInit McuDisableFlashInit McuDisableModeInit to allow integration with non-Autosar Application Initialization so that settings done by AppInit will not to be done again by MCU driver. See attached implementation from Quasar."
MCAL-21748	Bug	[MCU] Implement missing RCM ISR<*> Detailed description (how to reproduce it): Configure a reset source to also generate an interrupt by setting bits in RCM_SRIE. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Since no RCM ISR exists in "Mcu_RCM_Irq.c", there is no ISR available to register in IVT. When a reset source will generate an interrupt, the execution will jump to an unimplemented (thus escalating into a hard fault exception) or dummy interrupt handler (thus not allowing the application to gracefully shutdown the application before the system reset occurs , making the RCM_SRIE configuration completely useless). Expected behavior: RCM ISR exists in "Mcu_RCM_Irq.c" and is registered in IVT. When a reset source will generate an interrupt, the execution will jump to the aforementioned ISR and the application will be able to gracefully shutdown the application before system reset occurs. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Implement a RCM ISR in "Mcu_RCM_Irq.c".
MCAL-21749	Bug	[MCU] Add missing CMU_LOC reset source<*> Detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		# Configure and enable CMU0. # CMU0 indicates a loss of clock when FIRC goes below the programmed threshold. Preconditions: Platform contains CMU (i.e. bug can manifest only on S32K118). Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: System is reset with RCM_SRS[CMU_LOC] = 1 and RCM_SSRS[SCMU_LOC] = 1, but the "Mcu_GetResetReason" and "Mcu_GetResetRawValue" functions do not indicate a "CMU Loss-Of-Clock Reset". Expected behavior: System is reset with RCM_SRS[CMU_LOC] = 1 and RCM_SSRS[SCMU_LOC] = 1, and the "Mcu_GetResetReason" and "Mcu_GetResetRawValue" functions do indicate a "CMU Loss-Of-Clock Reset". Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Add missing CMU_LOC reset source
MCAL-21753	New	New Feature [SPI] Update the driver follow new MCL implement ,,"NewWorkDescription: Update the driver follow new MCL implement Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Update the driver follow new MCL implement"
MCAL-21754	Bug	[LIN] Abnormal SYNC byte<*> Detailed description (how to reproduce it): Customer is facing an issue when they get a 0x00 on the bus in place of the SYNC byte (0x55) (see the snapshot in the attachment). The issue doesn't happen on every transmission. It is random. Currently LIN driver writing the transmit FIFO with DATA[FRETSC] set and DATA[FRETSC] cleared (not DATA[R9T9] clear as RM said) to send a break character. Seemly it causes this issue because when customer using CTRL[SBK] to send a break character, they do not face this issue anymore. Please investigate and update the code. Developer: We have a problem when using CTRL[SBK] to send BREAK field. Please see the email in attachment file. So, we changed to using DATA[FRETSC]. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21759	Bug	[SPI] Missing sequence configuration in case of not using FLEXIO<*> Detailed description (how to reproduce it): Version: S32K14x_4.2_MCAL_1_0_1_RTM_P4_HF1 In Spi_Ipw_SyncTransmit function from Spi_IPW.c, sequence configurations are not correct when do not using FLEXIO: #if(defined FLEXIO_0_SPI_0_USED defined FLEXIO_0_SPI_1_USED) #if ((FLEXIO_0_SPI_0_USED == STD_ON) (FLEXIO_0_SPI_1_USED == STD_ON)) VAR(uint32, AUTOMATIC)SPI_REG_FLEXIO; / Get sequence configuration / / violates ref Spi_IPW_c_REF_6 Array indexing shall be the only allowed form of pointer arithmetic. / Ipw_pcSequenceConfig = &Spi_pcSpiConfigPtr->pcSequenceConfig[Sequence]; / Get the number of jobs in the sequence / Ipw_JobsCount = Ipw_pcSequenceConfig->NumJobs; Ipw_pcJobIndexList = Ipw_pcSequenceConfig->pcJobIndexList; #endif #endif Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21771	New	New Feature [MCU] Do not allow user configure number for string parameter ,,"NewWorkDescription: There is a problem with "RAM Section Size Linker Symbol", it has String type, so we can configure anything without any error. Example configure 0 value. The code will be generated like this and it will raise error in compilation step: extern VAR(uint32, MCU_VAR) 0[]; It will raise error during compilation. We should update to not allow user configure like that. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A"
MCAL-21785	New	New Feature [SPI] Has a problem with CPHA=0 and CS is continues when using FLEXIO ,,"NewWorkDescription: Has a problem with CPHA=0 and CS is continues on FLEXIO IPV Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Has a problem with CPHA=0 and CS is continues In RM, For only CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer. So Driver will remove support CS continues in CPHA=0"
MCAL-21808	Bug	[FLS] FLS_EXT_SECTOR_X sectors are not listed in epd files<*> Detailed description (how to reproduce it): In the Tresos GUI, the value FLS_EXT_SECTOR_x in the dropdown list is generated dynamically using the ID of the Fls sector (Fls_Sector_0 > FLS_EXT_SECTOR_0, Fls_Sector_1 > FLS_EXT_SECTOR_1, and so on). The Fls-epd-file does not offer this option and a the corresponding FLS_EXT_SECTOR_x cannot be chosen. The reason is because epd file only have FLS_EXT_SECTOR_0. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: Epd files contain FLS_EXT_SECTOR_X option in FlsPhysicalSector. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A

MCAL-21811	Bug	[LIN] Frame identifiers 60 (0x3C) to 61 (0x3D) shall always use classic checksum<*>
		
		Detailed description (how to reproduce it):
		In Lin_LPUART.c, Lin_LPUART_RxBuffFullInterruptHandler function.
		u32Lin_DataReg =
		REG_READ32(LPUART_DATA_ADDR32(u8LPUART_Channel));
		case LIN_TX_SLAVE_RES_COMMAND:
		/ Store received data /
		au8LPUART_Buffer[u8LogicalChannel][u8BufferPtr] = (uint8)(u32Lin_DataReg & LPUART_DATA_8BITS_MASK_U32);
		/ The last byte has been received /
		if(u8BufferPtr == (u8BufferLength (uint8)1U))
		{

Table continues on the next page...

ID	Subtype	Headline and Description

		/ Update LIN channel frame operation status to LIN_RX_COMPLETE_STATE /
		Lin_au8LinChFrameStatus[u8LogicalChannel] = LIN_RX_COMPLETE_STATE;
		u8Lin_FrameID = au8LPUART_Buffer[u8LogicalChannel]
		[LPUART_PID_BYTE_OFFSET_U8];
		/ Checksum Error /
		switch(u8Lin_FrameID)
		{
		case LPUART_MASTER_REQUEST_DIAGNOSTIC_ID_U8:
		case LPUART_SLAVE_RESPONSE_DIAGNOSTIC_ID_U8:
		au8LPUART_Buffer[u8LogicalChannel]
		[Lin_Lpuart_au8BufferLength[u8LogicalChannel] ((uint8)1U)] =
		(uint8)Lin_LPUART_ChecksumCalc(u8LogicalChannel, LIN_CLASSIC_CS);
		break;
		default:
		au8LPUART_Buffer[u8LogicalChannel]
		[Lin_Lpuart_au8BufferLength[u8LogicalChannel] ((uint8)1U)] =
		(uint8)Lin_LPUART_ChecksumCalc(u8LogicalChannel, LIN_ENHANCED_CS);
		break;
		}
		u8Lin_Checksum = au8LPUART_Buffer[u8LogicalChannel]
		[Lin_Lpuart_au8BufferLength[u8LogicalChannel] ((uint8)1U)];
		if(u8Lin_Checksum != (uint32)(u32Lin_DataReg &
		LPUART_DATA_8BITS_MASK_U32))
		{
		/ Set LIN channel frame error status to LIN_CHECKSUM_ERROR /
		Lin_au8LinChFrameErrorStatus[u8LogicalChannel] =
		LIN_CHECKSUM_ERROR;
		}
		}
		else
		{
		SchM_Enter_Lin_LIN_EXCLUSIVE_AREA_18();
		{
		/ Enable Receiver Interrupt /
		/
		violates ref Lin_LPUART_c_REF_3 cast from unsigned int to pointer
		violates ref Lin_LPUART_c_REF_7 cast should not be performed
		/
		REG_BIT_SET32(LPUART_CTRL_ADDR32(u8LPUART_Channel),
		LPUART_CTRL_RIE_MASK_U32);
		}
		SchM_Exit_Lin_LIN_EXCLUSIVE_AREA_18();
		}
		/ Noise Error has occurred ? /
		/ violates ref Lin_LPUART_c_REF_3 cast from unsigned int to pointer /
		if(LPUART_DATA_NOISY_MASK_U32== (u32Lin_DataReg &
		LPUART_DATA_NOISY_MASK_U32))
		{
		/ Set LIN channel frame error status to LIN_NOISE_ERROR /
		Lin_au8LinChFrameErrorStatus[u8LogicalChannel] = LIN_NOISE_ERROR;
		}
		break;
		u8Lin_FrameID for slave response case is not 0x3D, it is 0x7D and because LPUART_SLAVE_RESPONSE_DIAGNOSTIC_ID_U8 is defined is 0x3D, so the enhanced check sum will be used instead of classic check sum.
		Preconditions:
		S32K14X_MCAL4_2_RTM_P4_HF1_1_0_1

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: there should be error when using Lin slave response, calculating for classic checksum Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21813	Bug	[ADC] PDB sequence error flags are not fully cleared in ISR<*> Detailed description (how to reproduce it): In Adc_Pdb_ChannelSequenceError(u8Unit) function (Adc_Pdb_Irq.c) for each equivalent PDB unit, only error flag of the first PDB channel is cleared. In case of multiple error flags are raised at the same time, they are not fully cleared and ISR of sequence error might be occurred as a forever loop. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Clear error flags of all channels of the PDB unit
MCAL-21824	Bug	[ADC] Lack of DoubleBufferingMoreThanOneChannel_Object comment in xdm file<*> Detailed description (how to reproduce it): DoubleBufferingMoreThanOneChannel_Object comment is missing in Adc.xdm so traceability matrix report marks it as uncovered requirement Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Add <! / implements DoubleBufferingMoreThanOneChannel_Object / > at the beginning of DoubleBufferingMoreThanOneChannel node
MCAL-21826	Bug	[ICU] IcuDmaChannelRef Multiplicity<*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Detailed description (how to reproduce it): IcuDMAChannelRef is optional in EB tresos configuration. However, it is mandatory in epd file because it have lower and uper multiplicity equal 1: <ECUC-CHOICE-REFERENCE-DEF UUID="ECUC:511ba006-b102-4e66-9dd1-73bb51bc88ab"> <SHORT-NAME>IcuDMAChannelRef</SHORT-NAME> <DESC> <L-2 L="EN"> Icu DMA Channel Reference Reference to the DMA Channel configure for the Request </L-2> </DESC> <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <ORIGIN>NXP</ORIGIN> <POST-BUILD-VARIANT-MULTIPLICITY>false</POST-BUILD-VARIANT-MULTIPLICITY> <POST-BUILD-VARIANT-VALUE>true</POST-BUILD-VARIANT-VALUE> <VALUE-CONFIG-CLASSES> <ECUC-VALUE-CONFIGURATION-CLASS> <CONFIG-CLASS>PRE-COMPILE</CONFIG-CLASS> <CONFIG-VARIANT>VARIANT-PRE-COMPILE</CONFIG-VARIANT> </ECUC-VALUE-CONFIGURATION-CLASS> <ECUC-VALUE-CONFIGURATION-CLASS> <CONFIG-CLASS>POST-BUILD</CONFIG-CLASS> <CONFIG-VARIANT>VARIANT-POST-BUILD</CONFIG-VARIANT> </ECUC-VALUE-CONFIGURATION-CLASS> </VALUE-CONFIG-CLASSES> <DESTINATION-REFS> <DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/TS_T40D2M10I1R0/Mcl/MclConfigSet/DMAChannel</DESTINATION-REF> </DESTINATION-REFS> </ECUC-CHOICE-REFERENCE-DEF> So, when customer does not use EB tresos to config. They use another tool which use epd file then they have to use DMA at all. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21829	Bug	[ADC] DoubleBufferingMoreThanOneChannel can't run if use one channel<*> Detailed description (how to reproduce it): When use DoubleBufferingMoreThanOneChannel can't run with one channel in group. Preconditions: Use DoubleBufferingMoreThanOneChannel and

Table continues on the next page...

ID	Subtype	Headline and Description
		AdcEnableDoubleBufferingOptimization. Group configuration has only one channel Test Case ID (internal TC that caught the defect) optional: Adc_TC_0089 Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Developer investigate
MCAL-21847	New	New Feature [MCU] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k144_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21848	New	New Feature [LIN] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the

Table continues on the next page...

ID	Subtype	Headline and Description
		IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21850	New	New Feature [I2C] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: NA"
MCAL-21851	New	New Feature [ADC] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21853	New	New Feature [PORT] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21854	New	New Feature [CRCU] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]"
MCAL-21857	New	New Feature [ICU] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21858	New	New Feature [FLS] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21859	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[MCL] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21861	New	New Feature [EEP] Add new derivatives for S32K14X platform ,,"NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k142_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-21884	Bug	[ADC] Continuous conversion without interrupt use AdcGroupInBacktoBackMode<*> Detailed description (how to reproduce it): When configuring a group is Continuous conversion without interrupt and

Table continues on the next page...

ID	Subtype	Headline and Description
		AdcGroupInBacktoBackMode must use AdcGroupUsesChannelDelays. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: Adc_TS_0181 Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21885	New	New Feature [ICU] Support nonAutosar STANDBY operation as requested on CPR-MCAL-880 ,,"NewWorkDescription: Add support for wakeup accross STANBY as requested by CPR-MCAL\-880 and implemented on HF branch of tickets : MCAL-20077|[HF][ICU] Add support for wakeup source reporting after STANDBY wakeup MCAL-19299|[HF] [ICU] Add support for wakeup source reporting after STANDBY wakeup Requirement source: CPR-MCAL\-879 (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [NA]"
MCAL-21887	New	New Feature [MCU] Repair compilation error when MCU_ENABLE_USER_MODE_SUPPORT is ON ,,"NewWorkDescription: The build fails when MCU_ENABLE_USER_MODE_SUPPORT == STD_ON. This issue was introduced by MCAL-21747. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Rename `MCU_DISABLE_RESET_INIT` to `MCU_DISABLE_RCM_INIT` in the function `Mcu_Ipw_Init`. Move the `#if (defined(MCU_DISABLE_<IPV>_INIT) && (STD_OFF == MCU_DISABLE_<IPV>_INIT))` expression above the `#ifdef MCU_ENABLE_USER_MODE_SUPPORT` expression in the files ""Mcu_PMC.h"", ""Mcu_RCM.h"", ""Mcu_SIM.h"", and ""Mcu_SMC.h""."
MCAL-21910	New	New Feature [I2C] Support CRC feature for I2C on S32K1 MCAL 4.2 ,,"NewWorkDescription:

Table continues on the next page...

ID	Subtype	Headline and Description
		Customer request to support CRC feature for I2C. Here is the use case explanation: NXP chip will be Slave in the I2C communication. Abbreviations St = Start Condition Sr = Repeated Start Condition Sp = Stop Condition AM = Acknowledged by Master NM = Not Acknowledged by Master AS = Acknowledged by Slave NS = Not Acknowledged by Slave Master to Slave communication M->S :: St 90 AS 00 AS 0C AS Sr 91 AS 90 Slave address 00 Actual Data 0C Calculated CRC from Master Once Data ""00"" received from Master , Slave shall send Ack for Data & Meanwhile shall calculate the CRC for received data. Once Master CRC ""0C"" received , Slave should compare the Master CRC ""0C"" with Slave Calculated CRC , if both CRC Matches, then Slave shall send AS (Ack), if not matched then NS (Nack). Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: N/A"
MCAL-21914	Bug	[SPI] Left and right operands are identical when do not configure any FLEXIO<*> Detailed description (how to reproduce it): When customer does not configure any FLEXIO. FLEXIO_0_SPI_0_H and FLEXIO_0_SPI_1_H are defined as 99u in Spi_Cfg.h. So the value of FLEXIO0_HW equal FLEXIO1_HW then compiler reports that warning: Warning [Pa134]: Left and right operands are identical. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Compiler warning. Expected behavior: No compiler warning. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...]
MCAL-21923	New	New Feature [MCU] Correct the findings of the code review against checklist ,,"NewWorkDescription: Correct the findings of the code review against checklist.

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Check the contents of the ""Detailed_Findings"" sheet in the ""Checklist_for_Code_Review_Intermediate.xlsx"" report added in MCAL-21525."
MCAL-21926	Bug	[PWM] Cannot configure difference clock point for difference FTM channels<*> Detailed description (how to reproduce it): Configure three Ftm module: FTM_2, FTM_3 and FTM_5. FTM_2 has clock reference point to McuClockReferencePoint_SIRCDIV1 FTM_3 and FTM_5 have clock reference point to McuClockReferencePoint_SYS Cannot generate code because of this error: [C:\EB\tresos\EB23\..\..\..\NXP\AUTOSAR \S32K14X_MCAL4_2_RTM_1_0_1\eclipse\plugins\Pwm_TS_T40D2M10I1R0/ generate_PB/src/Pwm_PBcfg.c (signed):274]: [Error in variant:] PwmMcuClockReferencePoint is not uniformity. PwmMcuClockReferencePoint must same for all channels (node) Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Error in generate when configure difference clock point for difference FTM module. Expected behavior: each FTM module can refer to difference clock point. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21932	Bug	[GPT] Code for FTM4 up to FTM7 is not generated in Gpt_PBcfg.c<*> Detailed description (how to reproduce it): There is an issue in the generated file "Gpt_PBCfg.c". In that file, at "GptChannelConfigSet", there are only 3 FTMs mapped wherever the micro controller has 5 FTMs. I read the "AUTOSAR_MCAL_GPT_UM," at 4.7.1.2, it wrote that there should be 3 FTMs too. However, when I check the form "GptHwConfiguration" of Gpt plugin in tresos Studio, there are 5 FTMs. For more details, please refer to the attached pictures. Preconditions: FTM4 or FTM5 or FTM6 or FTM7 is used in Gpt configuration Test Case ID (internal TC that caught the defect) optional: Gpt_TC_0007 (Gpt_TS_001) Observed behavior: See detailed description. Expected behavior: See detailed description. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution optional: We have fixed the issue by manually changing the file "ARM_S32K14X_s32k146_lqfp100.properties" under the path "tresos\plugins\Resource_TS_T40D2M10I1R0\resource\". Instead of Gpt.Num_Ftm_Hw_Modules:4 it should be: Gpt.Num_Ftm_Hw_Modules:6 On S32K146 the Num_Ftm_Hw_Modules shall be 6 (not 4). On S32K148 the Num_Ftm_Hw_Modules shall be 8 (not 4).
MCAL-21936	New	New Feature [ADC] Update requirement CPR-MCAL-859.adc for software trigger ,,"Detailed description (how to reproduce it): Can we update this requirement CPR-MCAL-791.adc: The ADC driver shall provide an optional configuration parameter for reducing the number of interrupts required for processing the conversions of Adc Groups that consist of only 1 channel and are configured as ADC_ACCESS_MODE_STREAMING. This parameter shall be available only when DMA transfer is used. When this feature is enabled, only one interrupt will be raised after the completion of all stream conversions (as configured by AdcStreamingNumSamples parameter). An additional interrupt to be raised after half of the stream is converted shall also be configurable. Become this requirement CPR-MCAL-859.adc: The ADC driver shall provide an optional configuration parameter for reducing the number of interrupts required for processing the conversions of Adc Groups and are configured as ADC_ACCESS_MODE_STREAMING. This parameter shall be available only when DMA transfer is used. When this feature is enabled, only one interrupt will be raised after the completion of all stream conversions (as configured by AdcStreamingNumSamples parameter). An additional interrupt to be raised after half of the stream is converted shall also be configurable. We will remove the conditional "only 1 channel" to support more than 1 channel. Of course, the limited channels can be supported will depend on each platform and we need to update driver for all of platforms which support CPR-MCAL-791.adc. Additional update is needed in case of software trigger condition. Preconditions: [...] Test Case ID (internal TC that caught the defect) optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]"
MCAL-21937	Bug	[ADC] Continous without interrupt doesn't work if AdcPdbPeriodContinuousMode = 0<*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Detailed description (how to reproduce it): If a group is configured with following attribute: Continuous conversion Without interrupt PDB backtoback only The group will not work if AdcPdbPeriodContinuousMode = 0 or not big enough to for all channels to convert before starting over to the next sample. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Add constraint to Adc.xdm: AdcPdbPeriodContinuousMode shall not be 0. (findings) AdcPdbPeriodContinuousMode cannot be configured in ONESHOT mode.
MCAL-21961	Bug	[SPI] Use FLEXIOSPI_0_HWUNIT for index of SPI_DMA interrupt<*> NewWorkDescription: Use FLEXIOSPI_0_HWUNIT for index of SPI_DMA interrupt Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: In Spi_FlexIO_IsrRxDma_Spi_x, need to replace FLEXIOSPI_x_OFFSET by FLEXIOSPI_x_HWUNIT follow latest improvement
MCAL-21987	Bug	[BASE] enable global interrupt before go to sleep<*> Detailed description (how to reproduce it): ASR standard require before enterring sleep sequence, global interrupt should be disable, so sleep sequence can process tightly. The problem that is our platforms have wakeup events which are interrupt events. If global interrupt is disable, chip cannot wakeup afterward. Therefore global interrupt should be enabled again before executing "wait" "wfi" instruction. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Chip cannot wakeup from sleep mode Expected behavior: chip can wakeup from sleep mode Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional:

Table continues on the next page...

ID	Subtype	Headline and Description
		Therefore global interrupt should be enabled again before executing "wait" "wfi" instruction.
MCAL-21968	Bug	[CAN] CanControllerTrcvDelayCompensationOffset should be in ns<*> Detailed description (how to reproduce it): The value of CanControllerTrcvDelayCompensationOffset in tresos should be in nanoseconds but it is in number of Protocol Engine (PE) Clock periods (directly written to register) Expected behavior: ECUC_Can_00480 (ASR 4.3): Specifies the Transceiver Delay Compensation Offset in ns. The value written in register should be in number of Protocol Engine (PE) Clock periods. Proposed solution optional: Configuration parameter should be in nanosecond to follow AUTOSAR spec. It can be converted to number of Protocol Engine (PE) Clock periods using the following formula in Can_PBcfg.c: [!VAR "TrcvDelayCompensation" = "num:i(num:i(num:i(CanControllerFdBaudrateConfig/ CanControllerTrcvDelayCompensationOffset) $OutCanClockFrequencyFromMcuFd) div 1000000000)"!]
MCAL-21969	Bug	[FLS] Mismatch between function declaration and definition<*> Detailed description (how to reproduce it): Function declaration and definition mismatch in following functions: Function Name Declaration Definition Fls_Flash_PageWrite static FUNC(void, FLS_CODE) Fls_Flash_PageWrite(CONST*(uint32, AUTOMATIC) u32TotalBytes); static FUNC(void, FLS_CODE) Fls_Flash_PageWrite(VAR*(uint32, AUTOMATIC) u32TotalBytes) Fls_Flash_SectorRead|FUNC(Fls_LLDReturnType, FLS_CODE) Fls_Flash_SectorRead(CONST(Fls_PhysicalSectorType, AUTOMATIC) ePhySector,CONST(Fls_AddressType, AUTOMATIC) u32SectorOffset,*CONST*(Fls_LengthType, AUTOMATIC) u32DataLength,P2VAR(uint8, AUTOMATIC, FLS_APPL_DATA) pDataPtr);| FUNC(Fls_LLDReturnType, FLS_CODE) Fls_Flash_SectorRead(CONST(Fls_PhysicalSectorType, AUTOMATIC) ePhySector,CONST(Fls_AddressType, AUTOMATIC) u32SectorOffset,*VAR*(Fls_LengthType, AUTOMATIC) u32DataLength,P2VAR(uint8, AUTOMATIC, FLS_APPL_DATA) pDataPtr) Fls_Flash_SectorCompare|FUNC(Fls_LLDReturnType, FLS_CODE) Fls_Flash_SectorCompare(CONST(Fls_PhysicalSectorType, AUTOMATIC) ePhySector,CONST(Fls_AddressType, AUTOMATIC) u32SectorOffset,*CONST*(Fls_LengthType, AUTOMATIC) u32DataLength,CONSTP2CONST(uint8, AUTOMATIC, FLS_APPL_CONST) pDataPtr);|FUNC(Fls_LLDReturnType, FLS_CODE) Fls_Flash_SectorCompare(CONST(Fls_PhysicalSectorType, AUTOMATIC) ePhySector,CONST(Fls_AddressType, AUTOMATIC) u32SectorOffset,*VAR*(Fls_LengthType, AUTOMATIC) u32DataLength,CONSTP2CONST(uint8, AUTOMATIC, FLS_APPL_CONST)

Table continues on the next page...

ID	Subtype	Headline and Description
		pDataPtr) Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-21990	Bug	[LIN] Check wake-up event during LIN channel initialization<*> Detailed description (how to reproduce it): LIN driver for S32K1XX (IPV_FLEXIO, IPV_LPUART) does not check wake-up event during LIN channel initialization according to LIN250 as below: [LIN250] [If wake-up is supported by hardware (i.e. LinChannelWakeUpSupport == true), during LIN channel initialization it shall be checked if there was a wake-up event on the specific LIN channel, (if supported by hardware). If a wake-up event has been detected, the wake-up shall directly be reported to the EcuM via EcuM_SetWakeupEvent call-back function.]() Preconditions: Missing requirement LIN250 Test Case ID (internal TC that caught the defect) optional: Lin_TC_0024 (Lin_TS_013) Observed behavior: LIN does not check wake-up event during LIN channel initialization Expected behavior: Correct driver Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Add a block to check wake-up event in Initialization function.
MCAL-21991	Bug	[SPI][FLEXIO] Need to disable interrupt bit tranfer using POLLING mode<*> Detailed description (how to reproduce it): The test did not exit to the loop when using Async and Polling mode Preconditions: Using Spi_Asynctransmit() function and POLLING mode Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Need to disable interrupt bit tranfer using POLLING mode

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-21993	Bug	[EEP] Missing else keyword in Eep_Eeprom_SetFlexRamFuncCmd<*> Detailed description (how to reproduce it): In function Eep_Eeprom_SetFlexRamFuncCmd it seems that the else keyword is missing from the if statement that checks if eepControlCode is set to EEP_SET_QUICK_WRITES_MODE. Preconditions: EEP_ENABLE_QUICK_WRITES_API == STD_ON Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Even if bCmdAllowed is set to FALSE in the first "true" parte of the if, it may be changed to TRUE in the else part. Expected behavior: There should be an else keyword or, if not necessary, removed the true part of the if completely. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Further investigate the issue and decide if "else" should be added.
MCAL-21994	Bug	[SPI][FLEXIO] Incorrect RX length in Synctransmit mode<*> Detailed description (how to reproduce it): Incorrect RX length in Synctransmit mode Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: Incorrect RX length in Synctransmit mode Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Incorrect RX length in Synctransmit mode
MCAL-21999	Bug	[PORT] About the description for PortPinModeChangeable<*> Detailed description (how to reproduce it): Currently the description for PortPinModeChangeable is described as bellow: "Parameter to indicate if the mode of a port pin is changeable during runtime.True: Port Pin mode changeable allowed.False: Port Pin mode changeable not permittedThe function for changing the pin modes is not supported by the safety implementation." That description mention to safety implementation. I discussed with Razvan and he confirmed that it is in corrected description. The description should be updated like this as Port_00134: Parameter to indicate if the mode is changeable on a port pin during runtime. True: Port Pin mode changeable allowed. False: Port Pin mode changeable not permitted. Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-22002	Bug	[ICU] Remove comma from file history section<*> Detailed description (how to reproduce it): Comma in file revision history brakes the M4 macro processing. Generated plugin will have history comments in files where comma is present in the section. Regular expression used in Notepad to check all files in repo is: ifdef\(`M4_SRC_KEEP_REVISION_HISTORY', `dnl DO NOT modify this M4 line!([^,]*,\s) Preconditions: None Test Case ID (internal TC that caught the defect) optional: None Observed behavior: History of file changes shows up in release Expected behavior: History is not visible in release plugin file. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Remove comma from file history version
MCAL-22006	Bug	[CAN] Diver does not execute the DET error CAN_E_DATALOST when using interrupt RX with FIFO<*> Detailed description (how to reproduce it): Diver does not execute the DET error CAN_E_DATALOST when using interrupt with FIFO. if (FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32 == (can_status & FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32)) \\ { \\ REG_WRITE32(FLEXCAN_IFLAG1(FLEXCAN_##FC##_OFFSET), (uint32) ((can_status & FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32) & FLEXCAN_IFLAG1_CONFIG_MASK_U32));\\ if ((NULL_PTR != Can_pCurrentConfig->ControlerDescriptors[CAN_FC##FC##_INDEX| ##FC##_INDEX].Can_RxFifoOverflowNotification) && (FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32 == (can_mask & FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32))) { \\ Can_pCurrentConfig->ControlerDescriptors[CAN_FC##FC##_INDEX].Can_RxFifoOverflowNotification(); \\ }

Table continues on the next page...

ID	Subtype	Headline and Description
		\ }\ Requirement ID: CAN395: If the development error detection for the Can module is enabled, the Can module shall raise the error CAN_E_DATALOST in case of "overwrite" or "overrun" event detection. Implementation Hint: The system designer shall assure that the runtime for message reception (interrupt driven or polling) correlates with the fasted possible reception in the system. Preconditions: Rx FIFO=ON RX Processing = INTERRUPT CANCODESIZEOPTIMIZATION = OFF Test Case ID (internal TC that caught the defect) optional: Can_TC_1010 Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...]
MCAL-22010	New	New Feature [SPI] Develop Spi_FlexIO_SyncTransmit_Fast() function for FLEXIO IP vault ,,"NewWorkDescription: Develop Spi_FlexIO_SyncTransmit_Fast() funtion for FLEXIO IP vault Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Develop Spi_FlexIO_SyncTransmit_Fast() funtion for FLEXIO IP vault"
MCAL-22013	New	New Feature [PORT] Add "UntouchedPortPin" list containing port pins that are meant to be ignored by Port driver - not initialized or configured in any way" ,,"NewWorkDescription: Add "UntouchedPortPin" list containing port pins that are meant to be ignored by Port driver not initialzied or configured in any way, even by NotUsedPortPin configuration Requirement source: Future version of cPRT (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: In the PortConfigSet container, add a new list called 'UntouchedPortPin'. The list will contain elements having 2 attributes: PortPinPcr and SIUL2Instance. Checks should be put in place that a pin does not appear in both the list of configured pins and list of not touched port pins. Pins that are added in NotTouchedPortPin list will be excluded from the list of pins that get written with

Table continues on the next page...

ID	Subtype	Headline and Description
		the default value in NotUsedPortPin container
MCAL-22015	New	New Feature [PORT] Support to setting input pins on high-Z state ,,"NewWorkDescription: S32K supports to set input pin to high-Z. however, our driver does not have this. A FAE asked to support this feature. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Bellow is the solution from Razvan: Extending the enum Port_PinDirectionType (from Port_Port_Ci_Types.h) with one new value for high-Z: typedef enum { PORT_PIN_DISABLED = 0, /**< brief No settings: the pin is not available. / PORT_PIN_IN, /**< brief Sets port pin as input. / PORT_PIN_OUT, /**< brief Sets port pin as output. / PORT_PIN_HIGH_Z /**< brief Sets port pin as high-Z. / } Port_PinDirectionType This way: The user can config in Tresos the High-Z value to be used by Port_Init() function. The user can call the Port_SetPinDIrection() with High-Z mode There is no need for new API to be added in Port "
MCAL-22074	Bug	[LIN] Update Lin_FlexIO_au8FrameOnGoing after disturbance frame<*> Detailed description (how to reproduce it): After a frame data was sent with an error (disturb to Lin bus to corrupt SDU of lin frame: LIN_TX_HEADER_ERROR, LIN_TX_ERROR) then a new frame was not sent Reason: Lin_FlexIO_au8FrameOnGoing variable was not updated when a frame was transmitted done, even error occurred. Preconditions: After frame data was sent with an error (disturb to Lin bus to corrupt SDU of lin frame), continue to send a new frame Test Case ID (internal TC that caught the defect) optional: tc_fnc_lin_00215 Observed behavior: A new frame was not sent Expected behavior: A new frame was sent successful Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Update Lin_FlexIO_au8FrameOnGoing variable after disturb frame
MCAL-22077	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[LIN] Redefine FLEXIO_CHANNEL in LIN and I2C drivers ,,"NewWorkDescription: Got an error when running the int test ""Re_define FLEXIO_CHANNEL in CCD_I2c_Cfg.h and Lin_Cfg.h files"" Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Remove FLEXIO_CHANNEL define in Lin or I2c."
MCAL-22080	Bug	[CAN] Cannot use CanSpecifiedRAMBlockSize feature when only a RAM was support<*> Detailed description (how to reproduce it): Each Can controller can configure MBDSR0 to selects the data size for the region R0 of message buffer allocated in RAM. So it is possible for customer to configure various CAN with an individual payload size. However, driver does not allow to configure like that because this error when enable CanSpecifiedRAMBlockSize: "when only a RAM was supported, don't use this function". Preconditions: ASR4.2 RTM1.0.1 HF7 Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Cannot configure separately the payload size for each can controller. Expected behavior: Allow user to configure separately the payload size for each can controller. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-22090	Bug	[ICU] Review and include Types package to Top in design EA<*> Detailed description (how to reproduce it): Import Types package into Top in design UML (the package is missing) Preconditions: None Test Case ID (internal TC that caught the defect) optional: None Observed behavior: Missing package Types Expected behavior: Types package available Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22104	New	New Feature [SPI] SPI does not use all of FIFO because FCR always is set to 0

Table continues on the next page...

ID	Subtype	Headline and Description
		,,"Detailed description (how to reproduce it): In Spi_RegOperations.m, the value for FCR always is generated with this value: (uint32)(((uint32)(0) << 16) ((uint32)(0))). It mean that FCR value always is 0. So FIFO depth of 4 is not utilized. Preconditions: S32K14X_MCAL4_0_RTM_1_0_3 Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: The FIFO depth of 4 is not utilized. Expected behavior: All of FIFO should be use to improve the transition. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A"
MCAL-22109	Bug	[SPI][FLEXIO] Build fail when FLEXIO is disable<*> Detailed description (how to reproduce it): When FLEXIO is disable, compiler has got some error Preconditions: Do not use FLEXIO Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Create a define to protect all files FLEXIO
MCAL-22112	Bug	[MCU] Range of McuPeripheralClockFrequency for FTM is incorrect<*> Detailed description (how to reproduce it): As reference manual, the maximum frequency for FlexTimer governed by SYS_CLK. In HSRUN mode, SYS_CLOCK can be configured up to 112Mhz. However, we cannot configure clock for FTM as 112Mhz because of this error from EB tresos: Value out of range: is "1.12E8" but must be ">= 0" and "<= 80000000". There are some others module which have the maximum frequency is governed by SYS_CLK, so need to check also for all of peripherals. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Cannot configure FTM clock as 112Mhz. Expected behavior: Update the range of FTM clock up to 112Mhz. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution optional: N/A
MCAL-22120	New	New Feature [I2C] I2c_LPI2C_SlaveInterruptProcessing function has cyclomatic complexity greater than 20 ,,"NewWorkDescription: I2c_LPI2C_SlaveInterruptProcessing function has cyclomatic complexity = 23, so it need to be split. Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: split I2c_LPI2C_SlaveInterruptProcessing function function"
MCAL-22129	Bug	[CAN] Bus Off interrupt can not trigger<*> Detailed description (how to reproduce it): Customer report an error with bus off as bellow: "When I debug with the bus off. It always can not run into interrupt. When I check the NXP datasheet, it seems we need to stop the interrupt which set the register of the FlexCAN_FreezeMode in file CAN_Flexcan.c . After I add the code below. it works(only 1 & 4 line is added):" SchM_Enter_Can_CAN_EXCLUSIVE_AREA_05(); REG_BIT_SET32(FLEXCAN_MCR(u8HwOffset), FLEXCAN_MCR_FRZ_U32); REG_BIT_SET32(FLEXCAN_MCR(u8HwOffset), FLEXCAN_MCR_HALT_U32); SchM_Exit_Can_CAN_EXCLUSIVE_AREA_05(); above exclusive areas are added in Can_FlexCan_GotoFreezeMode. Currently, the write operation on MCR register in FlexCAN_FreezeMode was not protected by any Exclusive_area, this could cause a unexpected behavior. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Cannot enter bus off. Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: As description in Exclusive Area and Integration Manual, the write operation on MCR register in FlexCAN_FreezeMode must be protected by CAN_EXCLUSIVE_AREA_03.
MCAL-22145	New	New Feature [I2C] Remove notification in Synctransmit function. ,,"NewWorkDescription: the Synctransmit function no need a notification. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: the Notification should remove from Syntransmit functions on both ipv LPI2C and FlexIO The SyncTransmit function no need a notification so it should be removed from driver. the SyncTransmit function only notification when have an error occur, it is true behavior, so we no need remove error notification in Synctransmit functions anymore."
MCAL-22214	New	New Feature [I2C] Review change according to errata document for RTM 1.0.2 ASR 4.2 ,,"NewWork Description: Please complete errata analysis for S32K148 Errata maskset 0N20V & S32K118 Errata maskset 0N97V according to the information provided in errata review template (Log sheet) in attachment. Requirement source: Errata for S32K148, maskset 0N20V, Rev. 25/Oct/2018: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?newTargetListUrl=%2Fsites%2Fmcal%2FShared%20Documents&viewpath=%2Fsites%2Fmcal%2FShared%20Documents%2FForms%2FAllItems%2Easpx&id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx%2FS32K%20Errata%20Sheets%2FS32K148_0N20V_25Oct2018%2Epdf&parent=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx%2FS32K%20Errata%20Sheets] S32K118 Errata maskset 0N97V, Rev. 07/Jan/2019: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?newTargetListUrl=%2Fsites%2Fmcal%2FShared%20Documents&viewpath=%2Fsites%2Fmcal%2FShared%20Documents%2FForms%2FAllItems%2Easpx&id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx%2FS32K%20Errata%20Sheets%2FS32K118_0N97V_Rev07Jan2019%2Epdf&parent=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx%2FS32K%20Errata%20Sheets] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The New Feature ticket for reviewing the errata list will be closed with Resolved. The result of the errata review shall be recorded as an attached filled template file, that will list the errata for all IPs of the module in scope. Naming convention of this report should be: SMCAL_Errata_review_<MDL>_S32K148 0N20V. The analysis result will specify if each erratum impacts the module implementation or not and also the ticket ids for software workaround implementation, if needed. New Bug ticket will be raised for adding the software workaround implementation, containing the following information: Headline: shall contain the Errata ID (e.g: [ADC] New errata e4186 implementation for ADC) Reporter (role): Developer/Designer Link option: Add the New Feature ticket used for review with selection of "is a dependency for"."

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-22244	New	New Feature [I2C] Update FlexIO channel according Mcl ,,"NewWorkDescription: Update I2c FlexIO channel according Mcl driver Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Update FlexIO channel"
MCAL-22245	New	New Feature [SPI] Integrative IPV FLEXIO for S32K14x_4.2 ,,"NewWorkDescription: New IPV FLEXIO SPI need to be integrated for S32K14x_4.2 Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: New IPV FLEXIO SPI need to be integrated for S32K14x_4.2"
MCAL-22246	Bug	[MCU] Incorrect description and default value of "FIRC Regulator Enable" configuration parameter ,,"Detailed description (how to reproduce it): The behaviour of "FIRC Regulator Enable" configuration parameter (McuFIRCRegulatorEnable) is as follows: "FIRC Regulator Enable" checkbox is unchecked than the regulator is disabled (FIRCREGOFF bit set to 1) "FIRC Regulator Enable" checkbox is checked than the regulator is enabled (FIRCREGOFF bit set to 0) This does not correspond to the parameter description in Tresos (xdm file) which is incorrectly written as follows: SCG_FIRCCSR[FIRCREGOFF] Fast IRC Regulator Enable 0 Fast IRC Regulator is disabled. 1 Fast IRC Regulator is enabled. Note: Implementation Specific Parameter. The checkbox does not correspond to FIRCREGOFF bit value but to its negated value. In case the FIRC regulator is disabled this leads 2.5% lower frequency than the nominal 48MHz. Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution optional: Modify the description of ""FIRC Regulator Enable"" configuration parameter as follows: Fast IRC Regulator Enable 0 (checkbox unchecked) Fast IRC Regulator is disabled (FIRCREGOFF=1) 1 (checkbox checked) Fast IRC Regulator is enabled (FIRCREGOFF=0) Note: Implementation Specific Parameter Also set the default value of ""FIRC Regulator Enable"" configuration parameter to true (regulator enabled) as it is a HW default reset value.
MCAL-22264	Bug	[GPT] Wrong define symbolic names of channels<*> Detailed description (how to reproduce it): Configure the name of channel in EB is Gpt_Channel_0 with ID = 0. But in the generated files Gpt_Cfg.h this channel is defined : #define GptConf_GptChannelConfiguration_Gpt_Channel_0 (0U) This should be : #define Gpt_Channel_0 (0U) Preconditions: None Test Case ID (internal TC that caught the defect) optional: None Observed behavior: None Expected behavior: None Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: None
MCAL-22284	Bug	[FEE] The description for Fee_Init should be Asynchronous<*> Detailed description (how to reproduce it): The brief for Fee_Init is incorrect. it notes that this function is Synchronous. See this brief from Fee.c: / brief Service to initialize the FEE module. details The function Fee_Init shall initialize the Flash EEPROM Emulation module. pre The FEE module' s environment shall not call the function Fee_Init shall during a running operation of the FEE module. note The function Autosar Service ID[hex]: 0x00. note Synchronous note Non Reentrant api implements Fee_Init_Activity / Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: N/A
MCAL-22292	New	New Feature [I2C][LPI2C] Implement support for SDAS/SCLS pins ,,"NewWorkDescription: The LPI2C driver now works only with the SDA & SCL pins. Additional pin configuration options must be added in order to make the driver work with separated Master and Slave pins (Master on SDA & SCL and Slave on SDAS & SCLS). Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Add two additional pin configuration options to the ""I2cPinConfiguration"" variable in the configuration schema."
MCAL-22293	Bug	[ADC] Wrong "INVALID" check in node AdcEnableDoubleBuffering ,,"Detailed description (how to reproduce it): when set up both AdcEnableDoubleBuffering and AdcEnableDoubleBufferingOptimization is true. driver error: AdcEnableDoubleBuffering can be enabled only if AdcEnableDoubleBufferingOptimization is enabled. it should error when AdcEnableDoubleBuffering is true and AdcEnableDoubleBufferingOptimization is false Preconditions: NA Test Case ID (internal TC that caught the defect) optional: test suite Adc_TS_026 Observed behavior: NA Expected behavior: test suite Adc_TS_026 generate pass Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Update Adc.xdm in driver
MCAL-22295	Bug	[GPT] GptHwChannel for LPIT ipv can't refer to GptHwConfiguration on S32K118 derivative<*> Detailed description (how to reproduce it): GptHwChannel for LPIT ipv cann't refer to GptHwConfiguration on S32K118. It causes errors at generate step. Preconditions: Derivative: S32K118. Use LPIT channel Test Case ID (internal TC that caught the defect) optional:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Observed behavior: N/A Expected behavior: Lpit channel must be refer to GptHwConfiguration Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Correspond Lpit channel for S32K118
MCAL-22312	New	New Feature [CSEC] Improve Async mode ,,"NewWorkDescription: Async timeout should be separate with sync timeout Requirement source: [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: [...]"
MCAL-22313	New	New Feature [OCU] Add new derivatives for S32K14X platform ,,"Detailed description (how to reproduce it): New word: Add new derivatives for S32K14X platform Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: N/A Expected behavior: N/A Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: - NewWorkDescription: There are 3 new derivatives that have to be supported in S32K14X releases: s32k142_lqfp48 s32k144_lqfp48 s32k148_lqfp100 The ""S32K1 Pin Out"" folder for all derivative can be found in: [https://nxp1.sharepoint.com/sites/mcal/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Fmcal%2FShared%20Documents%2FDocuments%2FPlatforms%2FS32K1xx] These new derivatives represent same die; no extra features/instances of IPs but with shorter number of pins. Please have a look on the PinOut files and determine if some instances of the IPs are removed compared with a larger package: s32k142_lqfp48 can be compared with s32k142_lqfp64 s32k144_lqfp48 can be compared with s32k144_lqfp64 s32k148_lqfp100 can be compared with s32k148_lqfp144 Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional:_ N/A"
MCAL-22320	Bug	[Wdg] Wrong define the symbolic name for GPT channel ID in Wdg_ConfigType<*> Detailed description (how to reproduce it): the generated files GPT_Cfg.h ,this channel is defined : #define GptChannelConfiguration_0 (0U) But in the generated files Wdg_Cfg.h,this channel is used as GptConf_GptChannelConfiguration_GptChannelConfiguration_0 So need to update file Wdg_pluginMacros.m. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA Note:* in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22321	Bug	[ADC] Define ADC_UNIT_x_PDB_ERR_ISR_USED may not be generated when using multi VS<*> Detailed description (how to reproduce it): in .h file driver use xdm element to define macro. If this element is postBuild, Config Variant is VariantPostBuild, and this element value are different on each VS, Driver will use element's value of last VS to define macro. therefore we can miss or get some wrong macro. ex: on S32K1XX generate_PC\include\Adc_Cfg.h line: 202 element: AdcGroupDefinition is postbuild if config group_0 have 2 channel on VS_0 and have 1 on VS_2. #define ADC_CFGSET_GROUP_0_CHANNELS 1 generate_PC\include\Adc_CfgDefines.h line: 338, 339 element: AdcHwUnitId, AdcPdbSettings, AdcPdbChannelSequenceErrorEnable are postbuild driver will not define macro if config AdcPdbChannelSequenceErrorEnable is true on VS_0 and is false on VS_2. if on VS_0 AdcHwUnitId is ADC0 and is ADC1 on VS_2 driver will define : #define ADC_UNIT_1_PDB_ERR_ISR_USED Preconditions: config multi variant Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22323	Bug	[EEP] EepPageAsynchBehaviorEn node has only supported "vclass="PreCompile" ,,"Detailed description (how to reproduce it): EepPageAsynchBehaviorEn node has only supported PreCompile vclass (not support both PostBuild and PreCompile vclass). Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: see in EB of this node : Value Config Class: PostBuild Expected behavior: Value Config Class: PreCompile Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Change from: ifelse(M4_XDM_AR_RELEASE_REVISION,!!ASR_REL_4_0_REV_0003##,!!dnl <icc:v class=""PostBuild"">VariantPostBuild</icc:v> <icc:v class=""PreCompile"">VariantPreCompile</icc:v> ##,!!dnl <icc:v vclass=""PostBuild"">VariantPostBuild</icc:v> <icc:v vclass=""PreCompile"">VariantPreCompile</icc:v> become: ifelse(M4_XDM_AR_RELEASE_REVISION,!!ASR_REL_4_0_REV_0003##,!!dnl <icc:v class=""PreCompile"">VariantPostBuild</icc:v> <icc:v class=""PreCompile"">VariantPreCompile</icc:v> ##,!!dnl <icc:v vclass=""PreCompile"">VariantPostBuild</icc:v> <icc:v vclass=""PreCompile"">VariantPreCompile</icc:v> ##)dnl
MCAL-22325	New	New Feature [FLS] Add M4 tags to guard Hyper Flash feature ,,"NewWorkDescription: Using M4 to remove code relate to Hyper flash feature Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Using M4 to remove code"
MCAL-22330	Bug	[ADC] incorrect result of group with config AdcEnableDoubleBuffering is true

Table continues on the next page...

ID	Subtype	Headline and Description
		and more than one channel<*> Detailed description (how to reproduce it): incorrect result of group with config AdcEnableDoubleBuffering is true and more than one channel. driver just convert only first channel and dma call Adc_Adc12bsarv2_DmaTransferComplete0 macro: ADC_DOUBLE_BUFFERING_MORE_THAN_ONE_CHANNEL (STD_ON) was not defined. Preconditions: adc group with config AdcEnableDoubleBuffering is true and more than one channel Test Case ID (internal TC that caught the defect) optional: Adc_TC_089 Adc_TS_025 Observed behavior: NA Expected behavior: test report pass suite Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: update Adc_CfgDefines.h from line 582, 583 to generate define for both 4.2 and 4.3(AdcConfigSet/AdcHwUnit). In case config only one HW and AdcHwUnitId is ADC1, update Adc_CfgDefines.h file line 582 to generate define for both 4.0 ([!FOR "Unit" = "0" TO "count(AdcConfigSet/*/AdcHwUnit/*)"!][!//) and 4.2, 4.3 ([!FOR "Unit" = "0" TO "count(AdcConfigSet/AdcHwUnit/*)"!][!//).
MCAL-22361	New	New Feature [ADC] Lack of checking conversion complete flags in all channels ,,"NewWorkDescription: Adc_Adc12bsarv2_EndGroupConversion() and Adc_Adc12bsarv2_ReadGroup only checks COCO bit of last channel, it doesn't check COCO bit of all channels before last channel. Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Add checking condition for COCO bit at all channels."
MCAL-22363	Bug	[SPI] Update For FlexIO Only About SpiCsContinous Case In SpiExternalDevice<*> Detailed description (how to reproduce it): Please see and check chapter: SPI Master belong to 1626 pages of RM-REV.9 "SPI master mode can be supported using two Timers, two Shifters and four Pins. Either CPHA=0 or CPHA=1 can be supported and transfers can be supported using the DMA controller. For CPHA=1, the select can remain asserted for multiple transfers and the timer status flag can be used to indicate the end of the transfer. The stop bit is used to guarantee a minimum of 1 clock cycle between the slave

Table continues on the next page...

ID	Subtype	Headline and Description
		select negating and before the next transfer. Writing to the transmit buffer by either core or DMA is used to initiate each transfer." Preconditions: Need to config HW_Unit = FLEXIO_0 or FLEXIO_1 on EBTRESOS Test Case ID (internal TC that caught the defect) optional: Spi_TC_0059.c Observed behavior: With driver spi on autosar 4.2 lacked checking CPHA=0, CPHA=1. "SpiCsContinous case" Expected behavior: When CPOL =0, CPHA=0 (TRAILING) <---->SpiCsContinous =FALSE. When CPOL =0, CPHA=1 (LEADING) <---->SpiCsContinous =TRUE. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: [...] Please check RM an show contents above _*Detailed description
MCAL-22387	Bug	[MCU] ENET peripheral clock frequency reference value is wrong<*> NewWorkDescription: ENET clock source can be taken from SPLLDIV1_CLK, SIRCDIV1_CLK, FIRCDIV1_CLK, SOSCDIV1_CLK. But McuPeripheralClockFrequency value alway take from SPLLDIV2_CLK, SIRCDIV2_CLK, FIRCDIV2_CLK, SOSCDIV2_CLK Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: McuPeripheralClockFrequency value alway take from SPLLDIV1_CLK, SIRCDIV1_CLK, FIRCDIV1_CLK, SOSCDIV1_CLK.
MCAL-22393	Bug	[ICU] Wrong size of the Icu_ValueType variable<*> Detailed description (how to reproduce it): When running icu module to catch timestamp with DMA, the value that DMA transmits into the buffer-array(where the timestamp values shall be placed) in the wrong order. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: NA Expected behavior: The DMA value transmitted to the buffer must be in the correct order Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-22394	New	New Feature [ADC] Lack of checking group is already converted with feature ADC_OPTIMIZE_ONESHOT_HW_TRIGGER ,,"NewWorkDescription: Lack of checking group is already converted with feature ADC_OPTIMIZE_ONESHOT_HW_TRIGGER. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Add Adc_aGroupStatus[Group].eAlreadyConverted = ADC_ALREADY_CONVERTED; in Adc_Adc12bsarv2_EndPartialConversion and Adc_Adc12bsarv2_EndDmaPartialConversion function with feature ADC_OPTIMIZE_ONESHOT_HW_TRIGGER is ON."
MCAL-22395	New	New Feature [ICU] Improve the ISR configuration for S32K118 ,,"NewWorkDescription: S32K11x does not support the ISR for each LPIT channel. But the configuration for each channel is configurable. It might cause miss-understanding for customer which ISR should be used. Requirement source: N/A (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: N/A"
MCAL-22414	Bug	[CAN] Fix file version check in platform S32K14X_42<*> Detailed description (how to reproduce it): run file_verchecking.py to find log Preconditions: NA Test Case ID (internal TC that caught the defect) optional: [...] Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22416	New	New Feature [DIO] INVALID-Range for DioPortId node update ,,"Detailed description (how to reproduce it): Range for DioPortId node is not correct should be error raise ""Value out of range....""

Table continues on the next page...

ID	Subtype	Headline and Description
		For example: If config DioPortId=5 then error raise Value out of range: is ""5"" but must be "">= 0"" and ""<= 4 Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: Dio's driver works not correct on EB Tresos. Expected behavior: Dio's driver works correct on EB Tresos. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: With the example above, we can fix in Dio_42.xdm file"
MCAL-22420	Bug	[ETH] Eth_ConfigSet is not extern in case VARIANT_NO := 0<*> Detailed description (how to reproduce it): Test case condition: VARIANT_NO := 0 and using VariantPostBuild EthConfigSet was declared in Eth_PBCfg.c but this variable was not be extern in Eth_Cfg.h Preconditions: Test manual Test Case ID (internal TC that caught the defect) optional: Eth_TS_md_06 Observed behavior: NA Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Extern Eth_ConfigSet in case VARIANT_NO:=0
MCAL-22431	New	New Feature [DIO] Sync-up the content of Reg_eSys_Gpio.h in both DIO and PORT in order to avoid MISRA error ,,"NewWorkDescription: There is a MISRA error in PORT driver comes from duplicating file name Reg_eSys_Gpio.h in DIO and PORT plugins but different content. The file in DIO plugin does not contains definition for GPIO_PIDR_ADDR32. The PC-LINT will parse from up to down in list of plugins and it will parse the Reg_eSys_Gpio.h in DIO plugin so it will make the error: ""*GPIO_PIDR_ADDR32 is undefined symbol*"" Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Sync-up the content of Reg_eSys_Gpio.h in DIO & PORT"
MCAL-22434	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[PORT] The Pull Enable node should be visible no matter direction of pin ,,"NewWorkDescription: The Pull Enable node should be visible no matter direction of pin Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Remove the Editable condition in PortPinPE node in xdm file"
MCAL-22443	Bug	[PORT] Unused pins are initialized to GPIO mode instead of disabled mode<*> Detailed description (how to reproduce it): Port_Init() function initializes all unused port pins to GPIO mode (PORT_PCRn[MUX] = 1) instead of disabled mode (PORT_PCRn[MUX] = 0). To get the lowest power consumption unused pins shall be initialized to disabled mode (as they are after reset). Preconditions: N/A Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: After Port_Init() all unused pins are configured to GPIO mode instead of disabled mode. Expected behavior: After Port_Init() all unused pins are configured to disabled mode. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: Initialize unused pins to disabled mode.
MCAL-22451	Bug	[I2C] All nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false<*> Detailed description (how to reproduce it): The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Therefore, all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false Please see log file attached. if you want to generate new log then run check_postBuildNode.py file in Plugins folder. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Expected behavior: all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22452	Bug	[SPI] All nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false<*> Detailed description (how to reproduce it): The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Therefore, all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false Please see log file attached. if you want to generate new log then run check_postBuildNode.py file in Plugins folder. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Expected behavior: all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22457	Bug	[ICU] All nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false<*> Detailed description (how to reproduce it): The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Therefore, all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false Please see log file attached. if you want to generate new log then run check_postBuildNode.py file in Plugins folder. Preconditions: NA Test Case ID (internal TC that caught the defect) optional: NA Observed behavior: The nodes value will be changed between variants if it has POSTBUILDVARIANTVALUE is true. The defines created with the value depending on those nodes will also be changed. Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		all nodes are using to create defines value in Cfg.h should be had POSTBUILDVARIANTVALUE is false Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: NA
MCAL-22461	New	New Feature [FLS] Add the checking for MclLmemEnableCacheApi node when Enabling FlsSynchronizeCache on EB ,,"NewWorkDescription: Driver will build false when enable FlsSynchronizeCache and don't enable MclLmemEnableCacheApi. Requirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution optional: Add check condition that MclLmemEnableCacheApi must be enabled when enable FlsSynchronizeCache on EB."
MCAL-22468	Bug	[CAN] Stop acknowledge error reset after entry to VLPS mode due to enabled CAN<*> Detailed description (how to reproduce it): Customer is using CAN driver with FlexCAN clocked by SOSC (CanClockFromBus parameter is false). Before going to low power mode (VLPS) customer is calling Can_SetControllerMode(CAN_T_STOP) to stop the FlexCAN controller (they tried also CAN_T_SLEEP but this does not help). After that customer is calling Mcu_SetMode(VLPS_mode) to enter VLPS low power mode. During VLPS entry SACKERR (Stop Acknowledge Error) reset is generated. If customer sets manually MDIS bit (disable FlexCAN controller) before VLPS than the issue no longer happens. It is AUTOSAR platform that our software development is based on, that means, all of the state of CAN module should be managed by COM/CANSM according to AUTOSAR standard. I am afraid that disable CAN module directly before entering VLPS will infringe AUTOSAR standard, so would you plan to patch this issue in the next MCAL release? Preconditions: CAN controller configured to use SOSC clock. Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: SACKERR reset generated on entry to VLPS. Expected behavior: No SACKERR reset generated on entry to VLPS. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional: As a workaround it is possible to configure CAN controller to use BUS clock instead of SOSC. Or to manually set the MCR[MDIS] bit of FlexCAN controller before going to VLPS and wait until FlexCAN is disabled (MCR[LPMACK] set). A possible fix could be to update Can_SetControllerMode (CAN_T_STOP or

Table continues on the next page...

ID	Subtype	Headline and Description
		CAN_T_SLEEP) to set the MDIS bit (i.e. disable FlexCAN controller).
MCAL-22484	Bug	[I2C][LPI2C] Race condition between consecutive transfers when using repeated start<*> Detailed description (how to reproduce it): Set I2c_RequestType.bRepeatedStart = TRUE. Perform two consecutive TX calls to \`I2c_SyncTransmit\` or a TX call to \`I2c_SyncTransmit\` followed by another call to \`I2c_AsyncTransmit\` (with the last call having I2c_RequestType.bRepeatedStart = FALSE, of course). Because \`I2c_LPI2C_MasterSyncSend\` does not wait for a frame to complete when I2c_LPI2C_abRepeatedStart = TRUE (as it doesn't wait for a STOP signal to be detected on the bus), when the next call to \`I2c_SyncTransmit\` or \`I2c_AsyncTransmit\` occurs, the function \`I2c_LPI2C_MasterSendAddress\` resets the FIFO and may discard the last byte of the previous transfer if it's still residing in the FIFO buffer (i.e. if the the hardware module didn't yet load the internal shifter with the pending byte in the buffer). Preconditions: Set I2c_RequestType.bRepeatedStart = TRUE. Perform two consecutive TX calls to \`I2c_SyncTransmit\` or a TX call to \`I2c_SyncTransmit\` followed by another call to \`I2c_AsyncTransmit\` (with the last call having I2c_RequestType.bRepeatedStart = FALSE, of course). Note: The first transfer must be a TX transfer for this bug to manifest. Note: The likelihood of encountering this issue grows as CPU_FREQ / I2C_BITRATE is higher and drops as CPU_FREQ / I2C_BITRATE is lower. Test Case ID (internal TC that caught the defect) optional: N/A Observed behavior: Second call to \`I2c_SyncTransmit\` or \`I2c_AsyncTransmit\` may discard the last byte of the previous transfer that is still residing in the FIFO. Expected behavior: Second call to \`I2c_SyncTransmit\` or \`I2c_AsyncTransmit\` shall wait for the last byte of the previous transfer to be loaded from the FIFO into the internal shifter of the hardware module before resetting the FIFOs. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution optional:

4.2 RTM 1.0.1

ID	Subtype	Headline and Description
MCAL-15026	New	New Feature [ADC] Implement data consistency mechanism (exclusive areas) NewWork Description: Review the implementation of data consistency mechanism. Fill in the attached template. The attached template will be stored in version control using the name AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls

Table continues on the next page...

ID	Subtype	Headline and Description
		All global variables/structures fields/registers that are only read shall be placed in "Variables or Register only read" sheet and global variables/structures fields/ registers that are used in other operation than read shall be placed in "Variables or Register Analysis" sheet. In "Matrix for access type on global variables/registers (r, w, rmw)" section shall be filled in the variables or register (varx, register x)and the type of access in a specific function (func x) or interrupt. In "Exclusive Area Matrix" section shall be defined and exclusive areas used and the exclusivity matrix between them (on the column where the function was defined, the associated exclusive area shall be also defined). The following guidelines shall be followed in order to check the data consistency mechanism: - Exclusive Area shall be used to protect global variables, structures fields, registers - Rules to be used in order to set up exclusive areas: Rule 1: All read-modify-write operations on global variables/structures fields shall be protected by an exclusive area, if another function/interrupt is updating that memory space. Rule 2: All write operations on global variables/structures fields shall be protected by an exclusive area, if that operations may interfere with a read-modify-write operations on that global variables/configuration structures fields. Rule 3: For reentrant functions, global variables/structures fields or local static variables read-modify-write operation shall be protected by an exclusive area Rule 4: All registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*), shall be protected by an exclusive area, if another function/interrupt is updating that register. Rule 5: All write operations on registers shall be protected by an exclusive area, if that operation may interfere with a read-modify-write operation on that register Rule 6: For reentrant functions, registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*) shall be protected by an exclusive area - Note 1: A variable shall be read only once in a function. If it is needed to be read more than once, the variable shall be saved into a local variable - Note 2: Exclusive Areas implementation does not necessary mean disable interrupts - Note 3: Exclusive Areas are associated with functions, therefore there should be only one enter and exit from an exclusive area in the code Expected behavior: Add in driver/doc/design/ the report AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls Implement all the exclusive areas as in the report Check IM chapter 5.1 Exclusive areas to be defined in BSW scheduler and update it according to the design Output expected: - AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls - Exclusive areas defined in driver code - Updated IM chapter 5.1 according to exclusive areas defined in the code Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Review the implementation of data consistency mechanism. Fill in the attached template. The attached template will be stored in version control using the name AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15078	New	New Feature [WDG] When window mode is disabled the wdg should be triggered at half of the timeout Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): When window mode is disabled the wdg should be triggered at half of the timeout. Preconditions: Window mode disabled Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): FAE Observed behavior (not applicable in case of new features): When window mode is disabled the wdg is triggered at 0.8 of timeout. Expected behavior: When window mode is disabled the wdg should be triggered at half of the wdg timeout for a more robust implementation. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): When window mode is disabled the wdg should be triggered at half of the wdg timeout for a more robust implementation. In wdg_Swt.c update: return (uint32)(Swt_pConfigPtr->Swt_u32Timeout - (Swt_pConfigPtr->Swt_u32Window>>1)); to if (window mode enabled) return (uint32)(Swt_pConfigPtr->Swt_u32Timeout - (Swt_pConfigPtr->Swt_u32Window>>1)); else return (uint32)(Swt_pConfigPtr->Swt_u32Timeout /2)
MCAL-15106	New	New Feature [PORT] Set default value of checkbox for User Mode support to False in Port_Template.xdm Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Set default value of checkbox for User Mode support to False in Port_Template.xdm Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): NA Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): cPRT - by default all non ASR features should be disabled (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Set default value of checkbox for User Mode support to False in Port_Template.xdm
MCAL-15178	New	New Feature [ADC] Improvement cyclomatic complextity Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): As the attachment, there are lot of function with high cyclomatic complexity. They should be analysis and reduce under 10. Preconditions: N/A Test Case ID (internal TC that caught the bug) - optional N/A Trigger (not applicable in case of new features): N/A Observed behavior (not applicable in case of new features): N/A Expected behavior: N/A Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15434	New	New Feature [MCL] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use

Table continues on the next page...

ID	Subtype	Headline and Description
		the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15805	New	New Feature [PORT] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15806	New	New Feature [DIO] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath">

Table continues on the next page...

ID	Subtype	Headline and Description
		<a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15808	New	New Feature [MCL] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15810	New	New Feature [FEE] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15814	New	New Feature [ETH] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing

Table continues on the next page...

ID	Subtype	Headline and Description
		validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15969	New	New Feature [DIO] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16371	New	New Feature [EEP] Analyze the impact of using the same resources by FLS/EEP/CSEC and update drivers Analyze the impacts of using the same resources by FLS/EEP/CSEC and

Table continues on the next page...

ID	Subtype	Headline and Description
		update drivers. Consider: protection scheme, FCCOB, CCIF, and other hardware features which might influence the 3 drivers.
MCAL-16419	New	New Feature [DIO] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16426	New	New Feature [ETH] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4)

Table continues on the next page...

ID	Subtype	Headline and Description
		shall be used when calling trusting functions with 4 parameters and with return
MCAL-16602	New	New Feature [SPI] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-17191	Bug	[ICU] Update exclusive area for all IPVs<*>Now, exclusive areas in some IPvs overlap each other.Need update them correctly<*>Consider following rules:<*>Rule 4.1 All read-modify-write operations on global variables/ structures fields shall be protected by an exclusive area, if another function/ interrupt is updating that memory space.<*>Rule 4.2 All write operations on global variables/structures fields shall be protected by an exclusive area, if that operations may interfere with a read-modify-write operations on that global variables/structures fields.<*>Rule 4.3 For reentrant functions, global variables/ structures fields or local static variables read-modify-write operation shall be protected by an exclusive area. <*>Rule 4.4 All registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*), shall be protected by an exclusive area, if another function/interrupt is updating that register. <*>Rule 4.5 All write operations on registers shall be protected by an exclusive area, if that operation may interfere with a read-modify-write operation on that register.<*>Rule 4.6 For reentrant functions, registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*) shall be protected by an exclusive area. <*>Refer to attachment for more details
MCAL-17266	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[WDGIF] Fix number element of array base on Wdg instances NewWork Description: Number elements of some arrays related number of Wdg instances. But, number elements of that arrays is fix so it is not stable. Now, It need to be updated, number elements will base on number of Wdg instance.
MCAL-17640	New	New Feature [FEE] Remove buffer alignment from driver code and tests Remove buffer alignment from driver code and tests for the GHS compiler.
MCAL-17851	New	New Feature [DIO] Fix code after review against checklist for S12ZVMC256 RTM 1.0.0 ASR 4.0 NewWork Description: Review code against checklist. Fill the review result to the checklist. The checklist template and coding guideline are enclosed in attachment. Requirement source: sMCAL Release criteria document version 5.1: http://compass.freescale.net/go/228798570 Coding guideline verion 5.0 date 19-Jul-2016: https://www.nxp.com/go/230979668 Code review checklist verion 4.0, date Nov-2016: https://www.nxp.com/go/230979668 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The result of this activity will be the code review checklist and this ticket is treated as ?platform specific?. Code Review checklists, both Intermediate and Final, should be added in GIT and attached to this ticket. New ticket will be raised for the code modifications resulted for the ?code review? activity. This ticket will be used to update code, will be analyzed for all platforms and changes will be integrated for all affected platforms.
MCAL-18009	New	New Feature [MCU] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ?Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False if any PREFAILED or FAILED event is found True if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check

Table continues on the next page...

ID	Subtype	Headline and Description
		(NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True if the event is found (the first one found) with the requested status False if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) (new function) Returns: True if the number of events logged so far exceeds the DEM event buffer size elements False if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18010	New	New Feature [LIN] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in Dem_stub.c (file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) (already exists in this file) It will parse all events stored so far. It will return: False if any PREFAILED or FAILED event is found True if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True if the event is found (the first one found) with the requested status False if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer))

Table continues on the next page...

ID	Subtype	Headline and Description
		3. Dem_ClearEvents(void) (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) (new function) Returns: True if the number of events logged so far exceeds the DEM event buffer size elements False if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18177	New	New Feature [I2C] Implement exclusive error management in .xdms An LPI2C configuration should not issue any FLEXIO-related error messages and the other way around. Please refer to [MCAL-18052|https://jira.sw.nxp.com/browse/MCAL-18052] for details.
MCAL-18194	New	New Feature [GPT] Implement the requirement PR-MCAL-3307 (Gpt_ChangeNextTimeoutValue API) Detailed description (how to reproduce it): The APIyGpt_ChangeNextTimeoutValue needs to be implemented conform with latest cPRT on the following IPs: - eMIOS - eTIMER (already implemented - need check) - FTM - PIT - STM - TIM16b (already implemented - need check)
MCAL-18208	Bug	[PWM] Update DET handling for reset counter APIs<*>Detailed description (how to reproduce it):<*>Pwm_ResetCounter is replaced byyPwm_ResetCounterEnable andyPwm_ResetCounterDisable<*>DET handling should be updated with new APIs<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Observed behavior:<*>DET handles old API<*>Expected behavior:<*>DET handles newyAPI<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>update DET handling

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18229	New	New Feature [I2C] Requirements for Slave functionality A new function needs to be introduced in I2C's API. The corresponding requirements have been attached in the file "_sMCAL_I2C_New_Standard_Requirements.htm_". Rationale: * According to the I2C-bus specification, subsection 3.1.6, one of the five conditions that lead to the generation of a NACK is: "The receiver is unable to receive or transmit because it is performing some real-time function and is not ready to start communication with the master" As it stands now, it is not possible to voluntarily halt the Slave's availability for a given period of time, except if externally resorting to interrupt masking/suspension, which will affect an entire application and which will not conform to the I2C specification, because the Slave will not generate NACKs during this busy-interval. Remarks: * The new requirements should take effect from the Hearst BETA 0.9.0 ASR 4.0 (included) onwards. This decision is such as to not impact the current ongoing release of Kinetis (_SMCAL_4.2_S32K14X_RTM_1.0.0_).
MCAL-18222	New	New Feature [LIN] The u16LinChannelWakeupSupport element from Lin_ChannelConfigType should has type of uint8 or boolean Detailed description (how to reproduce it): Theyu16LinChannelWakeupSupport and u16LinEnableInWaitModeOnSleep element from Lin_ChannelConfigType hold STD_ON/STD_OFF values only. So it should has type of uint8 or boolean Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: Correct type of variable Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The u16LinChannelWakeupSupport element from Lin_ChannelConfigType hold STD_ON/STD_OFF values only. So it should has type of uint8 or boolean
MCAL-18223	Bug	[ADC] Update sign file for make file<*>Detailed description (how to reproduce it):<*>There are some file which are removed but still be marked as sign file in Adc.mak<*>Preconditions:<*>N/A<*>Test Case ID (internal TC that caught the defect) - optional<*>N/A<*>Observed behavior:<*>N/A<*>Expected behavior:<*>N/A<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>Remove the unavailable file from Adc.mak

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18228	New	New Feature [I2C] Fix inconsistency of output paths between ASR 4.0 and ASR 4.2 on S32K14X The file "I2C_VersionCheck_Src_PB.m" gets copied: * To "generate_PB" on ASR 4.0 * To "generate_PB/src" on ASR 4.2
MCAL-18231	Bug	[PORT] Update list of sign files for S32K14X<*>Detailed description (how to reproduce it):<*>There are some file which are removed but still be marked as sign file in Adc.mak<*>Preconditions:<*>N/A<*>Test Case ID (internal TC that caught the defect) - optional<*>N/A<*>Observed behavior:<*>N/A<*>Expected behavior:<*>N/A<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>Remove the unavailable file from Port.mak
MCAL-18260	New	New Feature [ADC] Support external trigger source for Hearst on ADC Adc on Hearst support two kind of trigger source: internal trigger and external trigger 11 PP4 (ETRIG0) to ADC0 trigger input 10 PAD1 (ETRIG0) to ADC0 trigger input 01 PAD15 (ETRIG0) to ADC0 trigger input 00 TIM0 output compare channel 2 to ADC0trigger input (output compare function onpin remains active unless disabled in timer config register TIM0OCPD[OCPD2]=1) So it should be configurable for the trigger sources on ADC. In case of TIM0 out put compare channel 2 is selected as trigger source, one check should be added to ensure that no port pin configured as external trigger for ADC
MCAL-18307	New	New Feature [I2C] Implement I2C_StartListening() on S32K14X This new function was recently introduced into I2C's API on Hearst (MC9S12ZVC) to provide additional Slave functionality.
MCAL-18308	New	New Feature [I2C] Implement master transfer completion notifications h2. _*On behalf of KsSTER*_ KsSTER would like to have the possibility of being notified at the end of an asynchronous Master transfer instead of always having to poll with I2C_GetStatus() until the channel's status makes the transition to "I2C_CH_FINISHED". Since this feature is quite general, the following features should be implemented on every supported platform:

Table continues on the next page...

ID	Subtype	Headline and Description
		* A callback notification shall be raised at the end of an asynchronous Master-Transmission. * A callback notification shall be raised at the end of an asynchronous Master-Reception.
MCAL-18310	New	New Feature [I2C] Requirements for Master functionality The following requirements are associated with the implementation of the feature described in [MCAL-18308|https://jira.sw.nxp.com/browse/MCAL-18308] : * A callback notification shall be raised at the end of an asynchronous Master-Transmission. * A callback notification shall be raised at the end of an asynchronous Master-Reception.
MCAL-18346	Bug	[FEE] Update to fix deviations from the design checklist<*>Detailed description (how to reproduce it):<*>Update the FEE driver design and code to failed items of the design checklist (MCAL-18244):<*>1.Update design:<*>In the FEE_HLD unit interface update the prototype of Fee_GetRunTimeInfo, it should have void return andan additional Fee_ClusterGroupRuntimeInfoType parameter.<*>2.Update design and code:<*>Fee_ForeignBlockConfig should be changed to Fee_aForeignBlockConfig<*>y<*>Additionally, update some parameter name to match the coding checklist.<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Observed behavior:<*>Design checklist deviations observed.<*>Expected behavior:<*>There should be no design checklist deviations.<*>There is no impact for the Fee customers.<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>1.Update design:<*>In the FEE_HLD unit interface update the prototype of Fee_GetRunTimeInfo, it should have void return andan additional Fee_ClusterGroupRuntimeInfoType parameter.<*>2.Update design and code:<*>Fee_ForeignBlockConfig should be changed to Fee_aForeignBlockConfig
MCAL-18391	New	New Feature [LIN] The u16LinChannelWakeupSupport element from Lin_ChannelConfigType should has type of uint8 or boolean update history Detailed description (how to reproduce it): Theyu16LinChannelWakeupSupport and u16LinEnableInWaitModeOnSleep element from Lin_ChannelConfigType hold STD_ON/STD_OFF values only. So it should has type of uint8 or boolean Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: Correct type of variable Note: in the Expected behavior field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The u16LinChannelWakeupSupport element from Lin_ChannelConfigType hold STD_ON/STD_OFF values only. So it should has type of uint8 or boolean
MCAL-18392	New	New Feature [ADC] Allow Dem calls to happen from interrupt Any DEM reaction is asynchronous anyway, therefore there is no need to notify (if DEM notification is really needed) immediately. Such notification can be postponed until next main function or any other API call. See AAI_310 for more detail
MCAL-18482	New	New Feature [I2C] Implement findings of code review against checklist for Hearst RTM 1.0.0 ASR 4.0 Implement the findings in MCAL-18232. Refer to the "MCAL_I2C_Checklist_for_Code_Review_Intermediate" file for detailed information.
MCAL-18490	Bug	[LIN] fix misra<*># Detailed description (how to reproduce it): because Lin wakeup support element of Lin_StaticConfig_ChannelConfigType was changed to uint8 type but in Lin_Cfg.c define uint16 type so it is cause misra error. <*> Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>all of test case <*> Observed behavior: <*> need change uint16 of Lin wakeup support in Lin_Cfg.c to uint8 type <*> Expected behavior: <*>need change uint16 of Lin wakeup support in Lin_Cfg.c to uint8 type <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...] Detailed description (how to reproduce it): <*> because Lin wakeup support element ofyLin_StaticConfig_ChannelConfigType was changed to uint8 type but in Lin_Cfg.c define uint16 type so it is cause misra error. <*> Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>all of test case <*> Observed behavior: <*>need change uint16 of Lin wakeup support in Lin_Cfg.c to uint8 type to correct misra <*> Expected behavior: <*>need change uint16 of Lin wakeup support in Lin_Cfg.c to uint8 type to correct misra <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18542	New	New Feature [I2C] MSR shall be cleared on a spurious interrupt According to the *_PR-MCAL-3272.i2c_* requirement, on a spurious interrupt (i.e., if at least one of flag is not set), the interrupt status register shall be cleared and the ISR shall return immediately.
MCAL-18579	Bug	[SPI] Incorrect information of SPI Generated Files<*>Detailed description (how to reproduce it):<*>At chapter 3.2 SPI Generated Files in Integration Manual, Spi_Cfg.c is still mentioned even though it is not available in MCAL SPI driver.<*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*>Spi_Cfg.c is still mentioned in Integration Manual<*> Expected behavior:<*>Remove Spi_Cfg.c information out of SPI Generated FIles in Integration Manual<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-18581	Bug	[DIO] Incorrect symbolic name for the channel group DioChannelGroup generated<*> Detailed description (how to reproduce it):<*> Following [SWS_BSW_00200] Symbolic Name values (_Symbolic Name Values in the implementation are using the short name of the Container in the ECUC prefixed with < ModuleAbbreviation>Conf_ (of the providing module) and the short name of the EcucParamConfContainerDef container [TPS_ECUC_02108]_), symbolic name of channel group should be defined as DioConf_DioChannelGroup_<short-name>. For example: DioConf_DioChannelGroup_DioChannelGroup_0. <*>Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*>Symbolic name for channel group is defined as: DioConf_DioChannelGroupIdentification_[!"node:name(.)"!]<*> Expected behavior:<*>Update to: DioConf_DioChannelGroup_[!"node:name(.)"!]<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-18590	New	New Feature [ADC] Analyse and fix GCC warnings for S32K14X Detailed description (how to reproduce it): Using Linaro compiller warning tests the following wornings are deteceted: array subscript is above array bounds [-Warray-bounds]@48 array subscript is above array bounds [-Warray-bounds]@44 for Adc.c file Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		Use Linaro (GCC) compiler Test Case ID (internal TC that caught the defect) - optional Adc_TS_CE01 Adc_TS_Eq_Cot_01 Adc_TS_M03 Adc_TS_M04 Adc_TS_M05 Observed behavior: array subscript is above array bounds [-Warray-bounds]@48 array subscript is above array bounds [-Warray-bounds]@44 warnings are detected Expected behavior: no warnings detected Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please analyse if these warnings can be removed without "commenting out" fro the RTM1.0.2 S32K release
MCAL-18597	New	New Feature [I2C] Files included in code templates shall be protected from multiple inclusion According to requirement "PR-MCAL-2672.i2c", files included in code templates shall be protected from multiple inclusion (an analogous mechanism is found in C header files). Every generated file shall be guarded by following a similar pattern: [!/* *** multiple inclusion protection *** */!] [!IF "not(var:defined(<file_name>_M'))"!] [!VAR "<file_name>_M"="'true'"!] ... rest of file ... [!ENDIF!]
MCAL-18614	Bug	[CAN] Wrong resource information for S32K146<*> Detailed description (how to reproduce it):<*> S32K146 have 3 controllers with support number of MBs in order are 32, 32, 16. And support CAN FD on CAN0 and CAN1.<*> Currently, the Can's resource has shown as below:<*> Can.CanConfigSet.CanController.NoMB:32,16,16<*>....<*> Can.CanConfigSet.CanFdEnableControllerA:STD_ON<*> Can.CanConfigSet.CanFdEnableControllerB:STD_OFF<*>Can.CanConfigSet.CanFdEnableControllerC:STD_OFF<*> Preconditions:<*>NA<*> Test Case ID (internal TC that caught the defect) - optional<*>NA<*> Observed behavior:<*>NA<*> Expected behavior:<*>NA<*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>yCheck and update correctly the Can resource for S32K146.
MCAL-18616	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[FLS] Remove Fls_Version.h file Detailed description (how to reproduce it): Remove Fls_Version.h file, include the AUTOSAR defines in the files which used to include Fsl_Version.h to have consistency between MCAL driver implementation. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18617	New	New Feature [FEE] Remove Fee_Version.h file Detailed description (how to reproduce it): Remove Fee_Version.h file, include the AUTOSAR defines in the files which used to include Fee_Version.h to have consistency between MCAL driver implementation. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: Fee_Version.h should be removed. Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Fee_Version.h should be removed.
MCAL-18618	New	New Feature [I2C] Guard the TDIE/RDIE enablement to maintain the integrity of a Slave channel Consider the following scenario: # SSR[FEF] = 1 # "I2C_LPI2C_GetStatus" is called, which clears SSR. Before "I2C_LPI2C_SlaveErrorHandler" is executed, a Slave interrupt occurs with SSR[AM0F] = 1, which calls "I2C_LPI2C_SlaveInterruptProcessing". # Before calling "I2C_LPI2C_SlaveEnableTransmitDataInterrupt" or "I2C_LPI2C_SlaveEnableReceiveDataInterrupt", another higher priority interrupt occurs, which executes the whole "I2C_LPI2C_GetStatus" function (now SIER[RDIE/TDIE] = 0 and I2C_aeChannelStatus = I2C_CH_ERROR_PRESENT). # The higher priority interrupt returns. # Now SIER[RDIE/TDIE] = 1 after resuming execution of the first interrupt.

Table continues on the next page...

ID	Subtype	Headline and Description
		Therefore, we have a Slave channel with "I2C_CH_ERROR_PRESENT" and SIER[RDIE/TDIE] = 1.
MCAL-18624	Bug	[MCU] Incorrect naming of Mcu Reset Type<*>Detailed description (how to reproduce it):<*> The reset type defined at chapter 8.2.4y Mcu_ResetType, AUTOSAR4.0.3 Specification of MCU Driver, is "MCU_SW_RESET" but in the configuration Mcu generated file, Mcu_Cfg.h, it is defined as "MCU_SOFTWARE_RESET". <*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> In ASR4.0.3, Mcu_ResetType is defined MCU_SW_RESET but in Mcu_Cfg.h, it is MCU_SOFTWARE_RESET.<*> Expected behavior:<*> Update MCU_SOFTWARE_RESET in Mcu_Cfg.h to MCU_SW_RESET<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-18638	Bug	[WDG] The Gpt timer may be stop wrongly<*> Imported from CQ ticket ENGR00395371.<*> The version 'BLN_SMCAL_4.0_CALYPSO_RTM_1.0.4' is not available in JIRA, 'DUMMY_VERSION' used instead.<*> Problem detailed description (how to reproduce it):<*> Depending of the WD timeout configured at Tresos and the execution rate of the function that requests the WD refresh, the WD driver can request to stops the refreshment without a real timing issue.<*> Issue scenario:<*> Observed on MPC574XG_MCAL4_0_RTM_1_0_4 and earlier.<*> From background, the function Wdg_ChannelSetTriggerCondition( ) is executed to request or not the next watchdog refresh using the Gpt_StopTimer().<*> The Wdg_ChannelSetTriggerCondition( ) gets the time elapse from the last watchdog refresh using the Gpt_GetTimeElapse (it s done in an isr from a hw timer) and evaluates against a variable Wdg_au16Timeout. <*> The variable Wdg_au16Timeout it s also recalculated inside of the function Wdg_ChannelTrigger( ). This function is executed at the hw s timer isr, the one that refresh the watchdog. <*> If the Wdg_ChannelTrigger( ) is executed in the middle of the Wdg_ChannelSetTriggerCondition( ), just after get the time elapsed and before enters at the exclusive area<*> Depending of the times configured at Tresos and the execution rate of our function, the if can stop the timer and don t refresh the watchdog anymore.<*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger: Observed behavior:<*>The GPT timer is incorrectly stopped<*>Expected behavior:<*>The GPT timer is not stopped<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional): <*> CE's comment: Gpt_GetTimeElapse should be put in the same ciritical section with the if condition so that it should not be interrupted by GPT ISR or

Table continues on the next page...

ID	Subtype	Headline and Description
		Wdg_ChannelTrigger
MCAL-18639	Bug	[PWM] Fix editable and invalid attribute in PwmFtmChannelFaultSettings in Pwm.xdm<*> Detailed description (how to reproduce it):<*>* For both release:<*> In some EDITABLE attribute, the expression is not fully implemented, led to some case, for example, the user sets PwmFaultSupport on,yPwmDisableOutputOnFault0 on, then fills in notification function name, when types invalid name, an error occurs to order user type vaild name. The user changes his mind and unchecksyPwmFaultSupport option, not to use fault notification, however, instead of that error disappearing, it still remains and user can not build configuration.<*>* For 4.0 release: In Pwm.xdm file, code to implement INVALID attribute for PwmFaultNotification to check syntax for fault ISR name is:<*>< a:da name="INVALID" type="XPath"><*>< a:tst expr="normalize-space(.) = '"NULL_PTR"' or normalize-space(.) = 'NULL_PTR' or normalize-space(.) = '"NULL"' or normalize-space(.) = 'NULL' or text:match(normalize-space(.),'^[_a-zA-Z]+[_0-9a-zA-Z]*$')" false="Invalid name of the PwmFaultNotification. Must be a valid C function name, NULL_PTR, "NULL_PTR", NULL, or "NULL"."/><*>< a:tst expr="(normalize-space(.) = '"NULL_PTR"' or normalize-space(.) = 'NULL_PTR' or normalize-space(.) = '"NULL"' or normalize-space(.) = 'NULL') and (../../../../../../PwmGeneral/PwmFaultSupport = 'true') and ./../ PwmFtmFaultFunctionality != 'AUTO_FOR_ALL_CHANNELS'"<*> true="A valid, not NULL_PTR C function must be defined for Fault 0 notification."/><*></a:da><*> Since the AUTO_FOR_ALL_CHANNELS parameter has been removed from last release, and ISR is always generated when fault option is enabled, the second <a:tst/> can be removed.<*> Proposed solution (Optional):<*> For both release, fix EDITABLE attribute of all parameters containing in PwmFtmChannelFaultSettings node.<*> For 4.0 release, remove second <a:tst/> tab in INVALID attribute of each PwmFaultNotification node.
MCAL-18641	New	New Feature y[EEP] Remove Eep_Version.h file Detailed description (how to reproduce it): Remove Fls_Version.h file, include the AUTOSAR defines in the files which used to include Fsl_Version.h to have consistency between MCAL driver implementation. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18644	Bug	[ADC] Update the disable value for SC register on 146 derivative<*> Detailed description (how to reproduce it):<*> The disable value for SC register on 146 derivative should be 0x3F instead 0x1F<*> Preconditions:<*>N/A<*> Test Case ID (internal TC that caught the defect) - optional<*> Adc_TS_003<*> Observed behavior:<*>0x1F value still trigger ADC and convert<*> Expected behavior:<*> Disable value should be 0x3F<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>N/A
MCAL-18659	Bug	[SPI] Incorrect calculation for SpiTimeCs2Cs Parameter<*>Detailed description (how to reproduce it): <*> The calculation for *SPI_CCR[DBT]* is incorrect that makes time delay *SpiTimeCs2Cs* is wrongly generated. Regarding to S32K-RM, this delay is equal to (DBT + 2) cycles of the LPSPI functional clock divided by the PRESCALE configuration. In this case, LPSPI functional clock is 4MHzy (SIRCDIV2), PRESCALE=0 and SCKDIV = 38, therefore DBT bit field should be set to 2 if we want a 1us SpiTimeCs2Cs delay. However, in Spi_PBCfg.c, DBT is generated 3 instead: *((uint32)(3) << 8u) | /* TimeCs2Cs: Should=1000ns, Is=1000, Error=0.0% */*. <*>By measurement, if DBT set to 0x3 as generated by EB Tresos, captured delay is about 1.27us in oscilloscope. If changing it to 2, the delay is about 1us, which is correct according to S32K-RM. <*> Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*> Generated parameter CCR[DBT] is incorrect <*> Expected behavior: <*> Update DetermineOptimalTimeSettings calculation in Spi_RegOperations.m in order to generate a correct value of CCR[DBT] <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...]
MCAL-18660	New	New Feature [I2C] Create separate completion notification callbacks for Slave As it is now, for Master there are two separate completion notification callbacks, differentiated by transmission and reception ("I2CMasterTransmitCompleteNotification" and "I2CMasterReceiveCompleteNotification"). However, there is only a single completion notification callback for Slave

Table continues on the next page...

ID	Subtype	Headline and Description
		("I2CSlaveTransferCompleteNotification"). In order to maintain as much uniformity as possible between Master and Slave, the Slave should also have two separate callbacks ("I2CSlaveTransmitCompleteNotification" and "I2CSlaveReceiveCompleteNotification").
MCAL-18674	Bug	[GPT] Incorrect interrupt flag handling <*>Detailed description (how to reproduce it):<*>Gpt_LPit_ProcessCommonInterrupt() routine (Gpt_LPit.c) is responsible for clearing interrupt flags in LPIT_MSR register in the end of ISR. As these flags are w1c (write 1 to clear), clearing a single flag using REG_BIT_SET32 macro (read-modify-write) clears also other flags set. This can lead to an unprocessed interrupt(s) in case two or more LPIT channel interrupts are simultaneously pending. Use REG_WRITE32 instead.<*> Preconditions:<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> All interrupt flags are cleared within Gpt_LPit_ProcessCommonInterrupt() by REG_BIT_SET32 macro.<*>Expected behavior:<*> Clear one flag for one interrupt<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>Replace REG_BIT_SET32 by REG_WRITE32 in Gpt_LPit_ProcessCommonInterrupt()
MCAL-18676	Bug	[GPT] Incorrect implementing bFreezeEnable bit for LPIT<*>Detailed description (how to reproduce it):<*> To enable freeze LPIT, DBG_EN bit should be 0b. To disable freeze LPIT, DBG_EN bit should be 1b. But in driver, the implementation is not correct: write 1 to enable freeze and write 0 to disable this feature.<*> Preconditions:<*>Test Case ID (internal TC that caught the defect) - optional<*>Gpt_TC_0009<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional): Call REG_BIT_SET32 for Lpit freeeze disable and REG_BIT_CLEAR32 for Lpit freeze enable.
MCAL-18677	New	New Feature [I2C] Fix master callback compilation issue on FlexIO In the file "I2C_FlexIO.c", when "I2C_DMA-USED == STD_ON", there is an additional paranthesis when calling "I2C_MASTER_RECEIVE_COMPLETE_NOTIFICATION" in the "I2C_FlexIO_InterruptProcessing" function.
MCAL-18678	Bug	[FLS] There is an inconsistency of the xdm class attribute for the flash sector parameters <*>Detailed description (how to reproduce it):<*>A flash sector contains multiple parameters, which from a logical point of view they are all tied to a physical flash sectors. So it makes sense that all parameters, Autosar and Vendor specific should have the same class.<*> Taken into account that all Autosar sector specific parameters are Precompile, check and update if necessary the Non-Autosar parameters to use the same

Table continues on the next page...

ID	Subtype	Headline and Description
		class.<*> Preconditions:<*>NA<*> Test Case ID (internal TC that caught the defect) - optional<*>NA<*> Observed behavior:<*>Some sector specific parameters are Precompile while others are Postbuild.<*> Expected behavior:<*> All sector specific parameters should have the same class.<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*> Check the configuration class(Precompile or Postbuild) for all Autosar parameters.<*> Update the NonAutosar parameters class to the same class as the Autosar ones, if they share the same container or if they are logically connected.<*> Run VSMD report to check any Autosar violation.
MCAL-18683	Bug	[ICU] Fix wrong clear bit flag interrupt <*> Detailed description (how to reproduce it):<*> Some IPVs write bit 1 to bit position flag interrupt from Register Flag Interrupt.<*> If using functionsyREG_BIT_SET all flag interrupt will be clear.<*> Preconditions:<*>NA<*> Test Case ID (internal TC that caught the defect) - optional<*>NA<*> Observed behavior:<*> All flag interrupt will be clear when do clear for an channel<*>Expected behavior:<*> Only clear flag interrupt for channel corresponding<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*> Write mask to register flag interrupt
MCAL-18685	New	New Feature [LIN] Review all used REG_BIT_SET32 macro in driver and implement change if need *Detailed description (how to reproduce it):* Following to the ticket MCAL-18674 - [GPT] Incorrect interrupt flag handling, customer wants to review all used REG_BIT_SET32 macro in GPT driver to ensure that there would be no extra incorrect implementation. Please review for all IPs in the driver at once. y *Preconditions:* Multiple interrupts occurred at same time y *Observed behavior:* Using macro REG_BIT_SET to clear w1c bit fields results in clearing all active interrupts *Expected behavior:* Only treated interrupts should be cleared. Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): Use macro REG_WRITE with specific bit field set only. y
MCAL-18688	Bug	[I2C] An interrupt flag is not cleared if its corresponding interrupt enable bit is not set<*> Detailed description (how to reproduce it):<*># Initiate transfer.<*># Enable MIER[TDF].<*># Disable MIER[NDIE].<*># Slave generates a NACK. Now MSR[NDF] = 1.<*># MSR[TDF] = 1 will trigger Master's ISR. However, MSR[NDF] is spurious, because MIER[NDIE] = 0. However, the driver will not clear MSR[NDF].<*>Preconditions:<*> MIER[NDIE] = 0.<*> Test Case ID (internal TC that caught the defect) - optional<*>N/A<*> Observed behavior:<*> MSR[NDF] is not cleared.<*> Expected behavior:<*> MSR[NDF] is cleared.<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>* Clear MSR in the beginning of the Master's ISR and process only the the enabled interrupt flags.
MCAL-18692	Bug	[FLS] FLS_STOP_SEC_CODE could not be reached<*>Detailed description (how to reproduce it):<*> Fls_Flash.c opens code section in line 283, the section is stopped in line 342 only if FLASH_INVALIDATE_PREFETCH_BUFFERS == STD_ON.<*> As this configuration switch has to be configured as false the compiler mentions an error.<*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*>STOP_CODE_SECTION can be reached only if FLASH_INVALIDATE_PREFETCH_BUFFERS == STD_ON.<*> Expected behavior:<*>[...]<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional)
MCAL-18700	New	New Feature [CAN] Reduce Cyclomatic Complexity Detailed description (how to reproduce it): Should reduce cyclomatic with the function has cyclomatic bigger than 20 Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] Expected behavior: [...] Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Split the function to sub-function which has cyclomatic smaller than 20.
MCAL-18703	Bug	[I2C] Interrupt handler does not properly clear spurious interrupt status bits<*>Detailed description (how to reproduce it): <*> If we have the following case (one interrupt is enabled but 2 flags will be set - one of them is spurious): <*> IrqEnableRegister = 0b1000 <*>IrqStatusRegister = 0b1001 <*> In this case, if the handler look only if the IrqEnableRegistery != 0 in order to process the interrupts, then it will try to process also the interrupt that is spurious. The driver musty clear the flags for spurious interrupt(in the above case, the interrupt that has the flag on 0b0001). The REG_BIT_SET instruction was masking this issue. <*> Preconditions: <*>Spurious interrupt occurred <*> Test Case ID (internal TC that caught the defect) - optional <*> NA <*> Observed behavior: <*> Flag is not clear when spurious interrupt occured <*> Expected behavior: <*> Flag is cleared when spurious interrupt occured <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Update implementation to detect spurious interrupt and clear flag in this case
MCAL-18708	Bug	[I2C][FlexIO] Asynchronous channel clears the error and timer status flags of a synchronous channel<*> Detailed description (how to reproduce it):<*> Configure both FLEXIO0_0_1 and FLEXIO0_2_3 channels.<*> Start an asynchronous transfer on FLEXIO0_0_1 (or FLEXIO0_2_3).<*> Start a synchronous transfer on FLEXIO0_2_3 (or FLEXIO0_0_1).<*> Now you have a race condition between these two channels. If FLEXIO0_2_3 will have a flag set in either TIMSTAT or SHIFTERR and if an interrupt will be asserted on FLEXIO0_0_1 just before FLEXIO0_2_3 handles those flags, then FLEXIO0_0_1 will incorrectly clear FLEXIO0_2_3's flags, considering them spurious (because the corresponding interrupt enable bits associated to FLEXIO0_0_1 will not be set, being a synchronous transfer).<*>Preconditions:<*>One channel configured as synchronous and the other channel configured as asynchronous.<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Interrupt handler incorrectly clearing the status flags of a synchronous channel engaged in a synchronous transfer.<*> Expected behavior:<*> Interrupt handler must leave the status flags of a synchronous channel engaged in a synchronous transfer untouched.<*> Note: in the Expected behavior field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*> In the interrupt handler, check the status of the other channel. If its status is either "I2C_CH_SEND" or "I2C_CH_RECEIVE", then do not clear its corresponding status flags, because they should not be considered spurious.
MCAL-18712	Bug	[BASE] Missing contanst USE_I2C_MODULE, USE_CSEC_MODULE and USE_EEP_MODULE check<*> Detailed description (how to reproduce it):<*> In Base_TS_T40D2M10I0R0\generate_PC\include\modules.h, the constant USE_I2C_MODULE used for other modules to check if I2C is present in the project andyUSE_EEP_MODULE used for other modules to check if EEP is present in the project are both missing. also missing for CSEC modules.<*> Preconditions: Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Missing constant check USE_I2C_MODULE and USE_EEP_MODULE.<*> Expected behavior:<*>[...]<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-18719	New	New Feature [GPT] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18880	Bug	[SPI][HF] The driver cannot transfer data in SPI_POLLING_MODE if use SpiDataShiftEdge=LEADING and select Continuous CS<*> Detailed description (how to reproduce it):<*> Spi driver has problem in SPI_POLLING_MODE without DMA or interrupt modeywithout DMA when use SpiDataShiftEdge=LEADING(Data is changed on the leading edge of SCK and captured on the following edge) in continue CS mode. WCF flag cannot set when write first frame into TDR register.<*> This issue occurs because the WCF flag is not working properly in the case of Master mode with SPI_POLLING_MODE without DMA or interrupt request without DMA and use SpiDataShiftEdge is LEADING and CS continuous.<*> So, the SPI driver cannot transfer successfully in that case.<*> Preconditions:<*>- use SPI_POLLING_MODE or interrupt mode (AsyncTransmit in non-DMA)<*>- SpiDataShiftEdge=LEADING<*> - Select Continue CS<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*> Observed behavior:<*> Transfer cannot finish.<*> Expected behavior:<*> Transfer is successful<*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*> Use interrupt of TDF flag instead of WCF flag.
MCAL-18950	Bug	[ADC] Multiplicity is set to 0 for AdcChannelDelay in autosar EPD files<*> Detailed description (how to reproduce it): <*> The configuration parameter AdcChannelDelay can be configured via Tresos GUI, but cannot be accessed via AUTOSAR configuration tool (like DaVinci Configurator 5) due to multiplicity restrictions. In *.epd formatted files, both LOWER and UPPER Multiplicity for this node is set to 0, making it unavailable in AUTOSAR configuration tool: <*> yyy <SHORT-NAME>AdcChannelDelay</SHORT-NAME> <*> <DESC> <*> <L-2 L="EN"> Delay of the associated PDB pretrigger for this channel. If Back to Back mode is not used for this group, the user must ensure the difference between the delays of different channels are big enough to accomodate the Adc conversions without PDB channel sequence errors. If Back to Back mode is used for this group, only the delay for the first channel will be considered - the rest of the channels will be converted in back to back mode. The delay value have to be a increasing numbers for each sequence of 16.</L-2> <*>yyy </DESC> <*>yyy < LOWER-MULTIPLICITY>0</LOWER-MULTIPLICITY> <*>yyy <UPPER-MULTIPLICITY>0</UPPER-MULTIPLICITY> <*>Preconditions: <*>n/a <*> Test Case ID (internal TC that caught the defect) - optional <*>n/a <*> Observed behavior: <*> In EPD files, multiplicity of AdcChannelDelay is set to 0 <*>Expected behavior: <*> In EPD files, multiplicity of AdcChannelDelay should be set to 1 <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>n/a

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18952	Bug	[FLS] Wrong DET API ID for erase and write failed<*> Detailed description (how to reproduce it): <*>The requirement: <*> *_[SWS_Fls_00104]_* _ The function Fls_MainFunction shall set the job result to MEMIF_JOB_FAILED and report the error code FLS_E_ERASE_FAILED to the DET if a flash erase job fails due to a hardware error. (SRS_BSW_00339, SRS_BSW_00385, SRS_BSW_00466)_ <*>_*[SWS_Fls_00105]* The function Fls_MainFunction shall set the job result to MEMIF_JOB_FAILED and report the error code FLS_E_WRITE_FAILED to the DET if a flash write job fails due to a hardware error._ <*> says that the FLS_E_ERASE_FAILED and FLS_E_WRITE_FAILED should be reported with FLS_MAINFUNCTION_ID. Currently, the first error reported at the beginning of Fls_Flash_SectorErase and Fls_Flash_SectorWrite, if the sectors are locked, are reported instead with FLS_ERASE_ID and FLS_WRITE_ID. <*> Preconditions: <*> Autosar 4.2, DET error reporting enabled, failed write/erase operation due to locked sector. <*> Test Case ID (internal TC that caught the defect) - optional <*> Fls_TC_01104(vnv) Fls_TC_01105(dev) <*> Observed behavior: <*> Requirements not fully implemented. <*> Expected behavior: <*>See above <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Replace FLS_ERASE_ID and FLS_WRITE_ID with FLS_MAINFUNCTION_ID. <*> See the attached email for more details.
MCAL-18953	Bug	[ICU] In the Port_Ci interrupt only the set flags should be ceared<*> Detailed description (how to reproduce it): <*> In functionyIcu_Port_Ci_ProcessInterrupt <*> After reading the interrupt flags, if a interrupt occurs on some other pins, there is a risk of loosing that event because all flags are cleared. <*> Preconditions: <*> User uses Port_Ci driver. <*> Test Case ID (internal TC that caught the defect) - optional <*> NA <*> Observed behavior: <*> Possible issue of losing interrupts because all flags are cleared instead of only the ones that are set. <*> Expected behavior: <*> Always read and clear the same flags <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...] Detailed description (how to reproduce it): <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Replacey <*>CONST(uint32, ICU_CONST) u32RegPort_CiISFR = REG_READ32(PORT_CI_ISFR_ADDR32(u8portNo)); <*>REG_WRITE32(PORT_CI_ISFR_ADDR32(u8portNo), 0xFFFFFFFFUL); <*>y <*>with <*>CONST(uint32, ICU_CONST) u32RegPort_CiISFR =

Table continues on the next page...

ID	Subtype	Headline and Description
		REG_READ32(PORT_CI_ISFR_ADDR32(u8portNo)); <*> REG_WRITE32(PORT_CI_ISFR_ADDR32(u8portNo), u32RegPort_CiISFR); <*>y <*>y
MCAL-18954	Bug	[SPI] Spi_Cancel causes execption if the Spi sequence has already completed<*> Detailed description (how to reproduce it): <*> Recently, my customer encountered one issue about AUTOSAR. the AUTOSAR version they are using is MPC5746R_MCAL4_0_RTM_1_0_2 <*> The customer usually calls Spi_Cancel() to cancel a SPI sequence if timeout occurs while waiting for the Sequence to be complete. <*> In slave mode, because the sequence is asynchronous with MCU software, it is possible that just before calling Spi_Cancel(), the SPI sequence to be canceled completes. In this case, there will be a memory access exception. <*> If the SPI sequence to be canceled completes just before calling Spi_Cancel(), the ISR of SPI (Non DMA mode) or DMA (DMA mode) will be processed before Spi_Cancel() is called. In this ISR, structure pointer pcJobConfig->pJobState->pAsyncCrtSequenceState() is assigned to NULL to finish the jobs in the sequence. However, the structure pointer is also accessed in the beginning of Spi_JobTransferFinished(), which is the subroutine of Spi_Cancel(). So after the software returns from ISR to Spi_Cancel() and calls subroutine Spi_JobTransferFinished(), there will be an exception because the pointer CPU wants to access has been cleared to NULL. <*>CE's comment: Some global variables are updated in Spi_JobTransferFinished, this function is asynchronous called in Spi_Canncel and SPI Interrupt/polling function but there is no exclusive area used to protect such variables. The job status and sequence status should be checked and continue the process only if it is still pending. <*>Preconditions: <*>- Spi is used in slave mode <*> - Spi slave sequence is started and cancelled. <*> - The Spi slave sequence completes before calling Spi_JobTransferFinished <*>Test Case ID (internal TC that caught the defect) - optional <*> Observed behavior: <*> The exception occurred inside Spi_JobTransferFinished function since the Job has been finished and pJobState-> pAsyncCrtSequenceState is NULL (this was updated in polling/interrupt of SPI) <*>Expected behavior: <*> No exception occurs <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional):
MCAL-18957	Bug	[FLS] Wrong path for FlsExternalSectorsConfigured when calculating CRC<*> Detailed description (how to reproduce it): <*> [When initializing the FLS module the CRC is computed differently at generation and at runtime. The problem is that at generation, the path for "FlsExternalSectorsConfigured" node is different form the one on Autosar 4.0.y] <*> Preconditions: <*>[N/A] <*> Test Case ID (internal TC that caught the defect) - optional <*>[N/A] <*> Observed behavior: <*>[At runtime the Fls_Init function fails.] <*> Expected behavior: <*>[The two CRC values should be equal.] <*> Note: in the Expected behavior field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> [An if-else guard should be used to make sure the path is correct for both 4.0 and 4.2 versions.]
MCAL-19167	New	New Feature [GPT] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-19201	Bug	[FLS] FlsMaxEraseBlankCheck infinite loop<*>Detailed description (how to reproduce it): <*> While performing an erase operation on an external sector, in asynchronous mode, if FlsEraseBlankCheck is activated andyFlsSectorSize is not a multiple of FlsMaxEraseBlankCheck then the driver gets stuck in an infinite loop. <*> Preconditions: <*> 1. External sector configured <*> 2. Erase function for that sector configured in asynchronous mode <*> 3.yFlsEraseBlankCheck activated <*> 4.yFlsSectorSize not a multiple ofyFlsMaxEraseBlankChecky <*> Test Case ID (internal TC that caught the defect)y <*> FLS_TS_00715 is a test suite created to catch this bug. <*> Observed behavior: <*> The driver remains stuck in an infinite loop, erasing the same sector all over again, since theyFls_Qspi_MainFunctionEraseBlankCheck function returns a success code but doesn't updateyFls_u32JobSectorIt in the given case. <*> Expected behavior: <*> TheyFls_Qspi_MainFunctionEraseBlankChecky function should update Fls_u32JobSectorIt so that the erase can continue with the next configured sector or end. <*> Proposed solution (Optional): <*> In theyFls_Qspi_MainFunctionEraseBlankChecky function, it should be checked only ifyFls_u32LLDNumberOfStepBack is bigger thanyFlsMaxEraseBlankCheck. On the else clause of this if the driver should end with succes and updateyFls_u32JobSectorIt.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-19211	Bug	[BASE] Moving typedef of pointerSizeType from Platform_Types.h to BASE specific header file<*> Detailed description (how to reproduce it): <*> In Platform_Types.h, there is a typedef for size of pointer type, pointerSizeType. However,y following ASR Specification of Platform Types, pointerSizeType is not listed at chapter 8.2 Type definitions. Also, in that same document, at chapter 7.1 General issues, it has mentioned: "[PLATFORM002] It is not allowed to add any extension to this file. Any extension invalidates the AUTOSAR conformity." <*> So far, DIO, CAN, FR modules have been using this typedef pointerSizeType so it would be better if we can move this one to the drivers specific header file. <*> Preconditions: <*>n/a <*> Test Case ID (internal TC that caught the defect) - optional <*>n/a <*> Observed behavior: <*>pointerSizeType defined in Platform_Types.h is not an ASR standard type so it should be moved to <driver> header file. <*> Expected behavior: <*> Move pointerSizeType from Platform_Types.h to <driver>.h <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> For ex:y <*>#if (CPU_TYPE == CPU_TYPE_64)y <*>typedef uint64 DioPointerSizeType;y <*>#elsey <*>typedef uint32 DioPointerSizeType;y <*>#endif
MCAL-19239	Bug	[MCU] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it): <*> Reported Baseline: <*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior: <*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>Spi_PBcfg[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!] <*>to: <*>Spi[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!]_PBcfg
MCAL-19240	Bug	[LIN] Mismatch between the name of generated variant files and information

Table continues on the next page...

ID	Subtype	Headline and Description
		within<*>Detailed description (how to reproduce it): <*> Reported Baseline: <*>The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*>Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior: <*>The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>Lin_PBcfg[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!] <*>to: <*> Lin[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!]_PBcfg
MCAL-19241	Bug	[SPI] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it): <*> Reported Baseline: <*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating SPI configuration with variant VS02, the generated file's name is:ySpi_VS02_PBcfg.c. However, in the content, it isySpi_PBcfg_VS02.c instead. Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior: <*>The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>Replace from: <*>Spi_PBcfg[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!] <*>to: <*>Spi[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!]_PBcfg
MCAL-19245	Bug	[PWM] Mismatch between the name of generated variant files and information within<*>Detailed description (how to reproduce it):<*>Reported Baseline:<*>The name of generated variant files is different from the one in the content of that file. <*>For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead.<*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Observed behavior:<*>Mismatch naming of the generated variant file and the one in the file. <*>Expected behavior:<*>The naming should be aligned. <*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-19247	Bug	[ICU] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*> Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-19248	Bug	[OCU] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*> Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):
MCAL-19249	Bug	[MCL] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*>Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*>

Table continues on the next page...

ID	Subtype	Headline and Description
		The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-19250	Bug	[FEE] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*> Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*>Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-19251	Bug	[FLS] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*>Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional)
MCAL-19257	Bug	[MCU] Guard the define USER_MODE_REG_PROT_ENABLED<*>Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define

Table continues on the next page...

ID	Subtype	Headline and Description
		USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19259	Bug	[SPI] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19260	Bug	[I2C] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19262	Bug	[DIO] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19263	Bug	[PORT] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-19264	Bug	[PWM] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19266	Bug	[GPT] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19267	Bug	[ICU] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19268	Bug	[OCU] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*> #define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19269	Bug	[MCL] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19275	Bug	[ETH] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19280	Bug	[ADC] Guard the define USER_MODE_REG_PROT_ENABLED<*> Original CQ Reported Baseline: BLN_IPV_CMU_SMCAL_4.0_01.16.00<*> Problem detailed description (how to reproduce it):<*> USER_MODE_REG_PROT_ENABLED is defined in several file without guard.<*> Should add ifndef USER_MODE_REG_PROT_ENABLED before define USER_MODE_REG_PROT_ENABLED<*> #ifndef USER_MODE_REG_PROT_ENABLED<*>#define USER_MODE_REG_PROT_ENABLED (ICU_USER_MODE_SOFT_LOCKING)<*>#endif
MCAL-19281	Bug	[OCU] S32K14x Code generation bug which leads to compile failure<*> Detailed description (how to reproduce it): <*> Due to [!IF "var:defined('postBuildVariant')"!]condition the size of the Ocu_InitHWMap_PB data is not generated when only one PB variant exists (or not variant has beed defined at top level). <*>This array is required to be passed as pointer to global configuration, data however due to the fact that the size is not generated this will result in the following error: <*>y <*>C:/Tools/EB/tresos/v21.0.0/workspace/IAS32K144_4.2/output/generated/src/Ocu_PBcfg.c", line 271: error #67: <*> expected a "}" <*>yyyyyyyy (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, (Ocu_ChannelType)255, <*> ^ <*>C:/Tools/EB/tresos/v21.0.0/workspace/IAS32K144_4.2/output/generated/src/Ocu_PBcfg.c", line 278: warning #12-D: <*> yyyyyyyy parsing restarts here after previous syntax error <*>yyyy }; <*>yyyy ^ <*> C:/Tools/EB/tresos/v21.0.0/workspace/IAS32K144_4.2/output/generated/src/Ocu_PBcfg.c", line 304: error #144: <*>yyyyyyyy a value of type "const Ocu_ChannelType *" cannot be used to <*>yyyyyyyy initialize an entity of type

Table continues on the next page...

ID	Subtype	Headline and Description
		"const Ocu_ChannelType (*)[]" <*>yyyy &Ocu_InitHWMap_PB <*>y <*>Preconditions: <*>Create only one PB variant and generate it <*>Test Case ID (internal TC that caught the defect) - optional <*>Observed behavior: <*>The compile issue describe above <*>Expected behavior: <*>Not much... just to compile... <*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>change this: <*>static CONST(Ocu_ChannelType, OCU_CONST)y Ocu_InitHWMap_PB[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][OCU_HW_CHANNELS_NO][!ENDIF!] <*>TOy this: <*>static CONST(Ocu_ChannelType, OCU_CONST)y Ocu_InitHWMap_PB[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!][OCU_HW_CHANNELS_NO] <*>y
MCAL-19283	Bug	[FLS] Wrong DET API ID for erase and write failed for treerunner<*>Detailed description (how to reproduce it): <*>The requirement: <*> *_[SWS_Fls_00104]_* _ The function Fls_MainFunction shall set the job result to MEMIF_JOB_FAILED and report the error code FLS_E_ERASE_FAILED to the DET if a flash erase job fails due to a hardware error. (SRS_BSW_00339, SRS_BSW_00385, SRS_BSW_00466)_ < *>_*[SWS_Fls_00105]* The function Fls_MainFunction shall set the job result to MEMIF_JOB_FAILED and report the error code FLS_E_WRITE_FAILED to the DET if a flash write job fails due to a hardware error._ <*> says that the FLS_E_ERASE_FAILED and FLS_E_WRITE_FAILED should be reported with FLS_MAINFUNCTION_ID. Currently, the first error reported at the beginning of Fls_Flash_SectorErase and Fls_Flash_SectorWrite, if the sectors are locked, are reported instead with FLS_ERASE_ID and FLS_WRITE_ID. <*>Preconditions: <*> Autosar 4.2, DET error reporting enabled, failed write/erase operation due to locked sector. <*> Test Case ID (internal TC that caught the defect) - optional <*>Fls_TC_01104(vnv) Fls_TC_01105(dev) <*> Observed behavior: <*> Requirements not fully implemented. <*> Expected behavior: <*>See above <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Replace FLS_ERASE_ID and FLS_WRITE_ID with FLS_MAINFUNCTION_ID. <*>See the attached email for more details.
MCAL-19429	Bug	[CAN] Moving typedef of pointerSizeType from Platform_Types.h to CAN specific header file<*> Detailed description (how to reproduce it): <*> In Platform_Types.h, there is a typedef for size of pointer type, pointerSizeType. However,y following ASR Specification of Platform Types, pointerSizeType is not listed at chapter 8.2 Type definitions. Also, in that same document, at chapter 7.1 General issues, it has mentioned: "[PLATFORM002] It is not allowed to add any extension to this file. Any extension invalidates the AUTOSAR conformity." So far, DIO, CAN, FR modules have been using this typedef pointerSizeType

Table continues on the next page...

ID	Subtype	Headline and Description
		so it would be better if we can move this one to the drivers specific header file. Preconditions: <*>n/a <*>Test Case ID (internal TC that caught the defect) - optional <*>n/a <*>Observed behavior: ;pointerSizeType defined in Platform_Types.h is not an ASR standard type so it should be moved to <driver> header file. <*> Expected behavior: <*>Move pointerSizeType from Platform_Types.h to <driver>.h <*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>For ex:y <*>#if (CPU_TYPE == CPU_TYPE_64)y <*>typedef uint64 DioPointerSizeType;y <*>#elsey <*>typedef uint32 DioPointerSizeType;y <*>#endif
MCAL-19474	Bug	[WDG] Incorrect description of WdgMaxTimeout in Wdg.xdm<*>Detailed description (how to reproduce it): <*> At the node Wdg/General/WdgMaxTimeout on Tresos GUI, Wdg Max Timeout is described: "The maximum timeout (miliseconds) to which the watchdog trigger condition can be initialized". However, the comment "milliseconds" is not correct due to the input value for this node is in "seconds" unit. <*>Preconditions: <*>n/a <*>Test Case ID (internal TC that caught the defect) - optional <*>n/a <*> Observed behavior: <*> Description of WdgMaxTimeout is incorrect with "milliseconds" unit <*> Expected behavior: <*> UpdateyWdgMaxTimeout description to "seconds" <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>n/a
MCAL-19475	Bug	[SPI] Spi generation error due to container naming<*>Detailed description (how to reproduce it): <*> Generation of Spi driver produces error messages like follows: <*> ERROR 18-02-08,11:41:23 (1806) Parsing file "~\Tresos\plugins \Spi_TS_T40D2M10I0R0/generate_PC/include/Spi_Cfg.h (signed)", line "514"y <*>...y <*> The XPath-expression "(node:value(SpiTransferWidth) >= 2)" caused an error: (1818) No value found for object ""y <*> ERROR 18-02-08,11:41:23 (2070) Failed to generate file "~\Tresos\plugins \Spi_TS_T40D2M10I0R0/generate_PC/include/Spi_Cfg.h (signed)"y <*> In the attachment you will find the configuration (Spi_gehtnicht.arxml) where the errors occur.y <*>y <*> Workaround: <*> Renaming all containers beneath SpiDriver container (like Spi_Channel,) by adding "_0" to the container name solved the issue (see attached configuration file Spi_geht.arxml). <*>y
MCAL-19478	Bug	[ECUM] Warnings are present in the Dem.xdm, visible when opening a project in Tresos<*>Detailed description (how to reproduce it): <*> The following warnings appear when opening a new project in Tresos with EcuM: <*> (File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/

Table continues on the next page...

ID	Subtype	Headline and Description
		EcuM.xdm, Line:102): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*> (File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:139): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*> (File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:159): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*>(File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:177): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*>(File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:623): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*>(File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:641): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*>(File:crypto:///C:/EB/tresos/23.0.0/plugins/EcuM_TS_T40D10M10I0R0/config/EcuM.xdm, Line:659): Invalid Attribute "IMPLEMENTATIONCONFIGCLASS" for tag "IMPLEMENTATIONCONFIGCLASS": No enum constant dreisoft.tresos.datamodel2.asc.attribute.ImplConfigClassAttribute.ConfigClass.PostBuildSelectable <*>Preconditions: <*>NA <*>Observed behavior: <*>Warnings reported <*>Expected behavior: <*>No warnings are reported <*>Proposed solution: <*>Remove the following attributes from EcuM.xdm file: <*><icc:v class="PostBuildSelectable">VariantPostBuildSelectable</icc:v>
MCAL-19479	Bug	[GPT] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*> Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. <*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*> Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-19558	Bug	[WDG] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it):<*> Reported Baseline:<*> The name of generated variant files is different from the one in the content of that file. <*> For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead.<*> Preconditions:<*>[...]<*> Test Case ID (internal TC that caught the defect) - optional<*>[...]<*> Observed behavior:<*>Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior:<*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*> Proposed solution (Optional):<*>[...]
MCAL-19693	Bug	[I2C] Implement interrupt functions for derivative S32K118<*> Detailed description (how to reproduce it): <*> For derivative S32K118, NVIC Interrupt ID of LPI2C Master & LPI2C Slave Interrupts have same value, instead 2 values like other derivatives. <*> Example: Please see the attachment files <*> Preconditions: <*>New derivative <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*>NA <*> Expected behavior: <*>Only 1 function interrupt for both LPI2C Master & LPI2C Slave Interrupts <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>1. Add a field in resource files: <*> # I2CUnifiedInterrupts : TRUE if the platform has only one interrupt vector for each LPI2C channel <*> I2CUnifiedInterrupts:TRUE <*>2. I2C_LPI2C_Irq.c
MCAL-19722	New	New Feature [PORT] Optimize the use of Port_Port_Ci_au16GPIODirChangeability[] array Detailed description (how to reproduce it): ArrayyPort_Siul2_au16GPIODirChangeability[] which is declared in Port_Siul2.c file is only used to perform a check when function Port_Siul2_SetPinDirection() is called, in order to see if the direction of the pin can be changed.yThe value of this check is requested from Port_SetPinDirection public API only when DET is on, as one can see in the piece of code below: #if (STD_ON == PORT_DEV_ERROR_DETECT)

Table continues on the next page...

ID	Subtype	Headline and Description
		ErrStatus = Port_Ipw_SetPinDirection(Pin, Direction, Port_pConfig); #else (void)Port_Ipw_SetPinDirection(Pin, Direction, Port_pConfig); #endif This means that the check performed in Port_Siul2_SetPinDirection() API is useless when DET is off, as it's result is anyway casted to void.y The scope of the ticket is to makeyPort_Siul2_au16GPIODirChangeability[] and the checks performed over it available only when DET is on. The ticket is already implemented in ASR 4.3 codebase, under the following JIRA link: https://jira.sw.nxp.com/browse/AMNG-956yPlease use this implementation as model. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: NA Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): MakeyPort_Siul2_au16GPIODirChangeability[] and the checks performed over it available only when DET is on
MCAL-19742	Bug	[I2C] Fix resource file for derivative s32k148<*> Detailed description (how to reproduce it): <*> Generate failed with derivative s32k148 <*>Preconditions: <*> Generate step <*> Test Case ID (internal TC that caught the defect) - optional <*>I2C_TS_M01 <*> Observed behavior: <*> Generate failed <*>Expected behavior: <*>Generate done <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): Change from: <*>I2C.I2CGlobalConfig.I2CChannel.I2CMasterConfiguration.I2C.LPI2C_0:2 <*>... <*>to: <*> I2C.I2CGlobalConfig.I2CChannel.I2CMasterConfiguration.LPI2C_0:2
MCAL-19753	Bug	[CAN] Implement interrupt functions for derivative S32K118<*> Detailed description (how to reproduce it): <*> For derivative S32K118, NVIC Interrupt ID ofyall Interrupts have same value, instead 2 values like other derivatives. <*> Example: Please see the attachment files <*>Preconditions: <*> New derivative <*> Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*>NA <*> Expected behavior: <*> Only 1 function interrupt for all Interrupts, which has same ID <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Creat two function support interupt handle for S32K118 <*>Can_ISR_NVIC_ID_10_FCA_SP <*>Can_ISR_NVIC_ID_11_FCA_SP
MCAL-19758	Bug	[LIN] Correct information for LinDevErrorDetect node<*> Detailed description (how to reproduce it): <*> 1)In Lin_42.xdm: <*> The label for LinDevErrorDetect node should be updated from "Lin Default Error Detection" to "Lin Development Error Detection". <*> The description for that node should be updated also from: <*> Switches the Default Error Detection and Notification ON or OFF <*>to: <*> Switches the Development Error Detection and Notification ON or OFF <*>2)In Lin_Cfg_42.h: <*> Update the brief also: <*>/** <*>* @brief Switches the Default Error Detection and Notification ON or OFF. <*>* <*>* @api <*>*/ <*>to: <*>/** <*>* @brief Switches the Development Error Detection and Notification ON or OFF. <*>* <*>* @api <*>*/ <*> 3) Update in the document for sync up <*>Preconditions: <*>Review <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*>[...] <*> Expected behavior: <*>[...] <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...]
MCAL-19759	Bug	[MCL] Update label for TrgMuxAdc0LockEn node<*> Detailed description (how to reproduce it): <*> Label for TrgMuxAdc0LockEn node should be update from: <*>< a:a name="LABEL" value="TrgMux ADC_1 Lock Enabled"/> <*>to: <*>< a:a name="LABEL" value="TrgMux ADC_0 Lock Enabled"/> <*> Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...]
MCAL-19763	New	New Feature [MCU] The FIRC, SOSC, and SPLL must be disabled by software in RUN mode before making transition to VLPR/VLPS Detailed description (how to reproduce it): This information come from Rev. 6, 12/2017: 26.4.4 VLPR/VLPS mode entry When entering VLPR/VLPS mode, the system clock should be SIRC. The FIRC, SOSC, and SPLL must be disabled by software in RUN mode before making any mode transition. In Mcu_SetMode: FIRC and SPLL are disabled, but SOSC is not. as the note from RM, SOSC should be disabled also before making transition to VLPS/ VLPR mode.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: NA Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): disable SOSC before making transition to VLPS/VLPR mode
MCAL-19767	Bug	[SPI] SpiByteSwap setting does not apply to SPI Slave channel<*>Detailed description (how to reproduce it): <*> Customer engineer raised: <*> - In the SpiExternalDevice configuration (in EB tresos): when SpiSlaveMode is set and SpiByteSwap is TRUE <*> ;- The generation code does not have the information for SpiByteSwap (generated by 'Spi_RegOperations.m' file) <*> /* LPspiDeviceAttributesConfig_PB0 Device Attribute Configuration of Spi*/ <*> static CONST(Spi_Ipw_DeviceAttributesConfigType, SPI_CONST) LPspiDeviceAttributesConfig_PB0[1] = <*> { <*>{ /* SpiExternalDevice_1 */ <*>(uint32)((LPSPI_TCR_CPOL_HIGH_U32 | /* Clock Polarity (Idle State) */ <*>LPSPI_TCR_CPHA_LEADING_U32 | (uint32)0u) & LPSPI_TCR_RESERVED_MASK_U32), <*>(uint32) 0x00000000u, <*>LPSPI_CFGR1_PCS0_IDLELOW_U32 /* Chip select polarity */ <*> | LPSPI_CFGR1_AUTOPCS_MASK_U32 | LPSPI_CFGR1_PINCFG_NORMAL_U32 | LPSPI_CFGR1_MATCFG_DIS_U32 <*>} <*>}; <*> Because all of above, the feature (byte swap) is not available for Slave mode for S32K1XX. <*> Developer's point of view: <*> - We detected this inconsistent between EB tresos configuration (SpiSlaveMode is set and SpiByteSwap) and meaning of these options with feature supported at driver. - {color:red} For now, driver is working as well with 32 bits SLAVE supported{color}, due to macro define is generated by getting data from resource, not from customer EB configuration. <*>- Without value of SpiByteSwap at EB config. <*> See the configuration in the attachment. <*> Preconditions: <*> In EB tresos, when SpiSlaveMode is set and SpiByteSwap is TRUE <*>Test Case ID (internal TC that caught the defect) - optional <*> NA <*>Observed behavior: <*>The generation code does not have the information for SpiByteSwap <*>Expected behavior: <*> The generation code have the information for SpiByteSwap (depend on user configuration data), make driver working under this condition. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>TBC

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20012	New	New Feature [ISELED] Create driver for S32K14X_4.2 Detailed description (how to reproduce it): N/A Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Observed behavior: N/A Expected behavior: N/A Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Create driver for S32K14X_4.2
MCAL-20013	New	New Feature [BASE] Add support for ISELED Detailed description (how to reproduce it): * Support for ISELED must be added in "Compiler_Cfg.h". * "Iseled_MemMap.h" must be created. * Support for ISELED must be added in "MemMap.h". Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Observed behavior: N/A Expected behavior: N/A Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): N/A
MCAL-20037	Bug	[CAN] Wrong interrupt functions for derivative S32K14X<*> Detailed description (how to reproduce it): <*> In function support for interupty ISR(Can_IsrFC##FC##_RxFifoEventsMbMix), Hardware Object for Transmit is processed from First TransmityMBy to MB 15 in the case RxFIFO andyCAN_RXFIFO_EVENT_UNIFIEDyare enable. <*>But in the case user config less than 15 HOH, it will be wrong when process Transmit hardware object. <*> Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: <*>[...] <*> Expected behavior: <*>[...] <*> Note: In the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Check the HOH transmit, ifythey are less than 15 MB, we will process them

Table continues on the next page...

ID	Subtype	Headline and Description
		from First MB TX index to MaxMB Count. <*>For example : <*> In the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Check the HOH transmit, ifythey are less than 15 MB, we will process them from First MB TX index to MaxMB Count. <*>For example : <*> if (0 < (Can_ControllerStatuses[CAN_FC##FC##_INDEXl ##FC##_INDEX].u8FirstTxMBIndex) < (uint8)(IdMax)) \ <*>\{ \ <*>Can_IPW_ProcessTx(CAN_FC##FC##_INDEX, \ <*>(Can_ControllerStatuses[CAN_FC##FC##_INDEXl ##FC##_INDEX].u8FirstTxMBIndex) , \ <*>(Can_pCurrentConfig->ControlerDescriptors[CAN_FC##FC##_INDEXl ##FC##_INDEX].u8MaxMBCount - (uint8)1U)); \ <*>} \ <*>else \ <*>\{ \ <*>Can_IPW_ProcessTx(CAN_FC##FC##_INDEX, \ <*>(Can_ControllerStatuses[CAN_FC##FC##_INDEXl ##FC##_INDEX].u8FirstTxMBIndex) , (uint8)(IdMax)); \ <*>} \
MCAL-20038	Bug	[LIN] Miss match in Lin_SendFrame function prototype<*> Detailed description (how to reproduce it): <*> I see there is a change in prototype of Lin_SendFrame from previous version (before RTM102). <*> The previous: FUNC(Std_ReturnType, LIN_CODE) Lin_SendFrame(VAR(uint8, AUTOMATIC) Channel, P2VAR(Lin_PduType, AUTOMATIC, LIN_APPL_DATA) PduInfoPtr) <*> The RTM102: FUNC(Std_ReturnType, LIN_CODE) Lin_SendFrame(VAR(uint8, AUTOMATIC) Channel, P2CONST(Lin_PduType, AUTOMATIC, LIN_APPL_CONST) PduInfoPtr) <*> CE's comment: According to the Autosar specification, the second parameter should be a pointer to variable. <*> Std_ReturnType Lin_SendFrame(uint8 Channel, Lin_PduType* PduInfoPtr) <*>Preconditions: <*> Test Case ID (internal TC that caught the defect) - optional <*>Observed behavior: <*> Compiler warning concerning the type of the second parameter. <*>Expected behavior: <*> No compiler warning concerning the type of the second parameter. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> The function prototype should be compliant with Autosar Specification
MCAL-20039	New	New Feature [PORT] Add support for package 48lqfp in generate file Detailed description (how to reproduce it): Missing adding support for package 48lqfp in Port_au16PinDescription array in generate file. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional Port_TS_008 Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		This array is generated for 48lqfp package with no element. Expected behavior: There are enough elements in this array for 48lqfp package Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update generate file
MCAL-20041	New	New Feature [SPI] Update driver support 16 bit and 32 bit in S32K14X_RTM_1.0.3 Detailed description (how to reproduce it): + When receive data in LPSPI driver then data is swaped and Dma can not transfer data from Src to TDR with 32 bit type so with 16 bit type ,data is swaped and 32 bit type , Seq is pending. Expected behavior: + In LPSPI.c . - Using Byeswap bit to swap data is true. - DMA transfers 16 bits twice to one frame 32 bit and fix position get data in RDR. Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-20046	Bug	[MCEM]S32K118 added support fault Double Bit Error Correction <*>We need add more function to handler hard fault when it happen for Double bit correction
MCAL-20064	Bug	y[WDG] For VariantPreCompile, there should be only one WdgModeConfig for each instance.<*>Detailed description (how to reproduce it): <*> For VariantPreCompile, not only one WdgModeConfig for each instance. <*>Preconditions: <*>Test Case ID (internal TC that caught the defect) - optional <*> Observed behavior: <*> incorrect syntax for the constraint "only one instance config for each instance of WDOG" <*> Expected behavior: <*> The condition check should work correctly. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> In Wdg.xdm <*>REPLACE: <*><a:tst expr="(../IMPLEMENTATION_CONFIG_VARIANT = 'VariantPreCompile') and ((count(/WdgInstance[.='WDOG0'])>1) or (count(/WdgInstance[.='WDOG1'])>1)or (count(/WdgInstance[.='WDOG2'])>1))" true="For VariantPreCompile, there should be only one WdgModeConfig for each instance."/> < *>BY: a:tst expr="(../IMPLEMENTATION_CONFIG_VARIANT = 'VariantPreCompile') and ((count(*/WdgInstance[.='WDOG0'])>1) or <*>(count(*/WdgInstance[.='WDOG1'])>1)or <*>(count(*/WdgInstance[.='WDOG2'])>1))" true="For VariantPreCompile, there should be only one WdgModeConfig for each instance."/>

Table continues on the next page...

ID	Subtype	Headline and Description

MCAL-20167	New	New Feature
		[DIO] Dio Plugin generation is slow (was Tresos Slows down after NCD update)
		Detailed description (how to reproduce it):
		the following test condition in Dio.xdm takes ~7min to evaluate
		<a:da name="DEFAULT" value="DioChannelGroup"/>
		<a:da name="INVALID" type="XPath">
		<a:tst expr="text:match(normalize-space(.),'^[_a-zA-Z]+[_0-9a-zA-Z]*$') or normalize-space(.) = ' ' " false="DioChannelGroupIdentification name should be entered as string."/>
		<a:tst expr="text:uniq(//DioChannelGroupIdentification, .)" false="DioChannelGroupIdentification is repeated for two or more channel groups.Please enter different symbolic name."/>
		</a:da>
		</v:var>
		It was verified that this is the root cause of the delays, by also commenting out these test conditions and testing.
		Reason for the slow down is a new NCD extract combined with a bad search from top level:
		//DioChannelGroupIdentification
		//WdgInstance
		Both xpath expressions are searching from the very top and thus taking huge time.
		Preconditions:
		- after NCD update: : a new ECU extract was imported. This increased from 61MB to 89MB (ARXML file size).
		Test Case ID (internal TC that caught the defect) - optional
		Observed behavior:
		- Dio Plugin generation is slow down
		Expected behavior:
		- Dio Plugin generation is not slow down
		Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
		Proposed solution (Optional):
		In Dio.xdm
		- REPLACE:
		<a:tst expr="text:uniq(//DioChannelGroupIdentification, .)" false="DioChannelGroupIdentification is repeated for two or more channel groups.Please enter different symbolic name."/>
		- BY:
		<a:tst expr="text:uniq(../../../../*/DioChannelGroup/*/DioChannelGroupIdentification, .)" false="DioChannelGroupIdentification is repeated for two or more channel groups.Please enter different symbolic name."/>

MCAL-20419	Bug	[LIN] Fix build error<*>Detailed description (how to reproduce it): <*> In Lin.h file, missing a ";": <*> FUNC(Std_ReturnType, LIN_CODE) Lin_SendFrame(VAR(uint8, AUTOMATIC) Channel, \ <*>P2VAR(Lin_PduType, AUTOMATIC, LIN_APPL_DATA) PduInfoPtr) <*> Preconditions: <*>Build <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: <*>Build failed <*> Expected behavior: <*>Build done <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>NA
MCAL-20453	Bug	[ETH] Unprotected access of array Eth_u8LockedTxBufCount<*>Detailed description (how to reproduce it): <*> In the file Eth_Enet.c , and function Eth_Enet_ReportTransmission(), the variable Eth_u8LockedTxBufCount has been read without protection in the below line: <*> for(u8BufIdx = 0U; (Eth_u8LockedTxBufCount[u8CtrlIdx] > 0U) && (u8BufIdx < u8TotalBufNum); u8BufIdx++) <*> This should be protected because this variable is accessed in Task OsTask_COMMUNICATION as well as in ISR Eth_TxIrqHdlr_1. <*> Preconditions: <*>Test Case ID (internal TC that caught the defect) - optional <*> Observed behavior: <*> Eth_u8LockedTxBufCount[u8CtrlIdx] > 0U) is not protected in exclusive area <*> Expected behavior: <*> Eth_u8LockedTxBufCount[u8CtrlIdx] > 0U) is protected in exclusive area <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional):
MCAL-20455	Bug	[SPI] SPI function "Spi_JobTransferFinished" reset due to NULL pointer<*> Detailed description (how to reproduce it): <*> The customer is encountering occassional RESET which was traced to an occurrence of a NULL pointer when referencing "pSequenceState" <*> within the "Spi_JobTransferFinished" function, that does not seem to be handled properly, possibly end of FIFO not detected/handled correctly. <*> See attached email from The customer on the details of this, and the modification of the SPI function to specifically check for this NULL pointer function in the <*> MCAL code which will avoid the earlier RESET issue. <*> I'm not sure of the Continental SPI transmission details that ends up with such a NULL pointer <*>Preconditions: <*>- Spi Slave is used <*> - Spi_Cancel function is used <*>Test Case ID (internal TC that caught the defect) - optional <*>Observed behavior: <*> - The driver accessed NULL pointer and caused reset <*>Expected behavior: <*> - The driver shall check the NULL pointer, and does not access to NULL pointer data <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): ;- Please see the customer proposal in attachment (MCAL - Spi_JobTransferFinished - NULL pointer (28 May 18).mht) <*>

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20460	New	New Feature [RTE] Add support for ISELED Detailed description (how to reproduce it): # Generate "SchM_Iseled.c" and "SchM_Iseled.h". # Add the previous files in "Rte.mak". Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Observed behavior: N/A Expected behavior: N/A Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): N/A
MCAL-20464	Bug	[FEE] Remove duplicated alignment for GHS compiler<*>Remove duplicateyAlign buffers to 64 bits for GHS compiler. It already implemented in the FLS module. See ticket:y-MCAL-17448-yfor more detail.
MCAL-20466	New	New Feature [GPT] Implement the requirement PR-MCAL-3307 (Gpt_ChangeNextTimeoutValue API) Detailed description (how to reproduce it): The APIyGpt_ChangeNextTimeoutValue needs to be implemented conform with latest cPRT on the following IPs: - eMIOS - eTIMER (already implemented - need check) - FTM - PIT - STM - TIM16b (already implemented - need check)
MCAL-20489	Bug	[Base] Code template must be signed<*>Detailed description (how to reproduce it): <*> Referring to 6.5.3. Commands from 2.4_Studio_documentation_developers_guide.pdf: <*> The following files must be signed: <*> Parameter definition files (.xdm) <*> All code templates and macro files (generally speaking all files that are processed by the code generator) <*> Due to Reg_eSys.h is moved to generate_PC and is processed by the code generator, Reg_eSys.h must be signed: [!SELECT "as:modconf('Resource')[1]"!][!// <*> [!IF "(string(text:split(node:value(node:fallback(ResourceGeneral/ResourceSubderivative,'')),'_')[1]) = 's32k118')"![!//

Table continues on the next page...

ID	Subtype	Headline and Description
		;#define PRAM_BASEADDR ((uint32)0x14000800) <*>[!ELSE!][!// <*>#define PRAM_BASEADDR ((uint32)0x14001000) <*>[!ENDIF!][!// [!ENDSELECT!][!// <*>Preconditions: <*>S32K RTM 1.0.3 Test Case ID (internal TC that caught the defect) - optional <*>N/A <*> Observed behavior: <*> The generate file is not signed <*> Expected behavior: <*> All generate file must be signed <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> In the Base.mak update SIGN_FILES as bellow: <*> SIGN_FILES = config/Base.xdm \ <*> generate_PC/include/modules.h \ <*>generate_PC/Reg_eSys.h
MCAL-20490	New	New Feature [RESOURCE] Update resource according to new Reference manual Rev.6 for S32K1XX RTM 1.0.1 ASR 4.2 NewWork Description: Please update resource according to latest Reference manual yto support the following derivatives * s32k148_lqfp144 * s32k148_lqfp176 * s32k148_mapbga100 * s32k146_lqfp64 * s32k146_lqfp144 * s32k146_lqfp100 * s32k146_mapbga100 * s32k144_lqfp100 * s32k144_lqfp64 * s32k144_mapbga100 * s32k142_lqfp100 * s32k142_lqfp64 * {color:#f79232}*s32k118_lqfp64 (Add new)*{color} * {color:#f79232}*s32k118_lqfp48*y*(Add new)*{color} * Note: The S32K14X 4.0 RTM 1.0.3 release will support to test successfully for S32K142, S32K144, S32K146, S32K148, S32K118 derivative. For S32K116 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K116. Requirement source: * S32K14x reference manual: Rev.6, 12/2017 (Attached atyMCAL-19303) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-20492	New	New Feature [GPT] Implement interrupt functions for derivative S32K118 Detailed description (how to reproduce it): The interrupt type of instance LPIT and FTM on derivative S32K11x and S32K14X is different.y

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: Interrupts of module GPT is failed on derivative S32k118 Expected behavior: Interrupts occur properly Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update Ipvault of LPIT and FTM
MCAL-20493	New	New Feature [PWM] Implement interrupt functions for derivative S32K118 Detailed description (how to reproduce it): The interrupt type of instance FTM on derivative S32K11x and S32K14X is different.y Observed behavior: Interrupts of module PWM is failed on derivative S32k118 Expected behavior: Interrupts occur properly Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update Ipvault of FTM
MCAL-20494	Bug	[SPI]UPDATE COMPILER WARNNING FOR SPI<*> Detailed description (how to reproduce it): <*> - When run some tests, i see that there are some warnnings when build test. <*> " Zero used undefined preprocessing identifierySPI_SWAP_DATA_SUPPORT_32BITS" <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> - add ifdefySPI_SWAP_DATA_SUPPORT_32BITS before checkySPI_SWAP_DATA_SUPPORT_32BITSy is ON or OFF.
MCAL-20496	Bug	[ICU] Warning when add module it EB Tresos<*> EB Tresos show warning when add ICU module. Message is: <*> The attributes postBuildVariantMultiplicity and multiplicityConfigClass may only be used for optional-, list- or choice-elements. Those attributes will be removed for element "IcuSignalNotification"!
MCAL-20497	New	New Feature [PORT] Replace error with warning in Tresos when configuring same functionality on more than 1 pin, as support for ISELED driver Detailed description (how to reproduce it): Please replace error with warning in Tresos when configuring same functionality on more than 1 pin, as support for ISELED driver ISELED driver is using Port plugin as a dependency. When working with

Table continues on the next page...

ID	Subtype	Headline and Description
		multiple strips, the same signal (eg SPI0_SCK) must be configured on more than one pin. Actually, the IoMuxing of this signal tells how many strips can be used on a SPI instance. If for example SPI0_SCK is available on 2 different PORT pins, it means that ISELED can use a max of 2 strips on the SPI0 interface. In the example above, at a given moment during runtime of the app, the SPI0_SCK signal will either be muxed on one of the 2 possible pins or on the other. This is done by ISELED driver using Port_SetPinMode() function. But, in order for the ISELED to know which mode it should use to configure the pins as SPI0_SCK, this information must be taken from Port configuration. This is why it is necessary for Port configuration to allow the user to configure same signal on more than one pin.y Instead of raising an error, the Port plugin should display a warning to the user when same signal is configured on more than one pin. y The proposed implementation is available in the Port repo, on the branchydbg_nxa19458_support_for_iseled, under tagyDBG_NXA19458_ADD_SUPPORT_FOR_ISELED y Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: NA Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The proposed implementation is available in the Port repo, on the branchydbg_nxa19458_support_for_iseled, under tagyDBG_NXA19458_ADD_SUPPORT_FOR_ISELED
MCAL-20498	Bug	[GPT] Incorrect GptChannelTickValueMax value<*>Problem detailed description (how to reproduce it): <*>GptChannelTickValueMax is maximum value which will be wrote to counter registers, and it is dependent on HW specific resolutions. <*>For 16-bit, 32-bit timers, GptChannelTickValueMax should be 65535, 4294967295 respectively. In general, for n-bit timers, GptChannelTickValueMax should be 2^n - 1. At the moment GptChannelTickValueMax is defined as 2 ^ n. <*>GTM channels' resolution is 16-bit timer, so GptChannelTickValueMax should be 65535. <*>EMIOS is 24-bit timer <*>FTM, LPTMR is 16-bit timer <*>STM, PIT, LPIT, RTC are 32-bit timer. <*>Preconditions: N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Trigger: <*>N/A <*>Observed behavior: <*>At the moment, GptChannelTickValueMax is always set as 2^n for n-bit timers for all platforms in EB Tresos. This might lead to a misunderstanding for users that they are able to set 2^n ticks to a GPT channel's period. <*>Expected behavior: <*>Correcting the GptChannelTickValueMax corresponding to a specific HW. <*>Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Change GptChannelTickValueMax in Gpt.xdm
MCAL-20500	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[ADC] Update requirement CPR-MCAL-791.adc Detailed description (how to reproduce it): Can we update this requirement CPR-MCAL-791.adc: The ADC driver shall provide an optional configuration parameter for reducing the number of interrupts required for processing the conversions of Adc Groups that consist of only 1 channel and are configured as ADC_ACCESS_MODE_STREAMING. This parameter shall be available only when DMA transfer is used. When this feature is enabled, only one interrupt will be raised after the completion of all stream conversions (as configured by AdcStreamingNumSamples parameter). An additional interrupt to be raised after half of the stream is converted shall also be configurable. Become: The ADC driver shall provide an optional configuration parameter for reducing the number of interrupts required for processing the conversions of Adc Groups and are configured as ADC_ACCESS_MODE_STREAMING. This parameter shall be available only when DMA transfer is used. When this feature is enabled, only one interrupt will be raised after the completion of all stream conversions (as configured by AdcStreamingNumSamples parameter). An additional interrupt to be raised after half of the stream is converted shall also be configurable. We will remove the conditional only 1 channel to support more than 1 channel. Of course, the limited channels can be supported will depend on each platform and we need to update driver for all of platforms which support CPR-MCAL-791.adc. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-20503	Bug	[PORT] PortPinMode can only be set to GPIO<*>Detailed description (how to reproduce it): <*> In Port epd-files the list of values for PortPinMode is incomplete and consists only of GPIO. <*> Preconditions: <*> - Non EB Tresos configuration tool is used (e.g: Vector Configuration tool that uses EPD files) <*> Test Case ID (internal TC that caught the defect) - optional <*> Observed behavior: <*> - The pin modes should be available so that they could be selected for appropriate pins. <*> Expected behavior: <*> - Only GPIO mode can be selected. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20509	New	New Feature [GPT] For 4.2 define GPT_CHANGE_NEXT_TIMEOUT_VALUE in "Gpt_Cfg_42.h" Detailed description (how to reproduce it): The APIyGpt_ChangeNextTimeoutValue needs to be implemented conform with latest cPRT on the following IPs: - eMIOS - eTIMER (already implemented - need check) - FTM - PIT - STM - TIM16b (already implemented - need check)
MCAL-20512	Bug	[ADC] ADC_Init wrong when config Adc logical diffrence with HW index<*>Detailed description (how to reproduce it): <*> When ADC config with more than one HW Unit, array Adc_HwUnitCfg gennerate folow HW index, but in driver code, we use Adc logical id to read data, so if HW index # logical id, configuration of HW Unit was swapped <*> Preconditions: <*> In AdcHwUnit we use 2 HW Unit: <*> Index Name Adc Hardware Unit Adc Logical id <*>0 AdcHwUnit_0 ADC0 1 <*>1 AdcHwUnit_1 ADC1 0 <*> Data for Adc_HwUnitCfg was genarated follow: <*>{Data config for AdcHwUnit_0,Data config for AdcHwUnit_1 } <*> That isn't right, it must be: <*> {Data config for AdcHwUnit_1,Data config for AdcHwUnit_0 } <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*>NA <*>Expected behavior: <*>NA <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Data for Adc_HwUnitCfg must genarate follow Adc logical id
MCAL-20560	Bug	[ICU] FTM module can be used by 2 drivers at once<*> Detailed description (how to reproduce it): <*>ICU_FTM_x_USED are not defined for the used modules. <*> The check done in MCL is based on those defines. <*> There are cases when PWM and ICU can use same FTM module and channel with no error on generate or build. The runtime code will not work. <*> Preconditions: <*> Use PWM and ICU drivers on same FTM module. Test Case ID (internal TC that caught the defect) - optional ; use attached test project <*> Observed behavior: <*>as in description - same module and channel used for PWM and for ICU. <*> Expected behavior: <*>Build should rise an error. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>generate the defines. <*>

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20831	Bug	[ETH] Update driver to avoid impact from supporting both ASR4.2 and ASR4.3<*> Detailed description (how to reproduce it): <*> There are some impacts when implemented asr4.3 <*> Preconditions: <*>NA <*> Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*> Some code implementation are working well in ASR4.3 but not in ASR4.2 <*>Expected behavior: <*> Code implementation support all ASR42, 43 and 40 <*> Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>[...]
MCAL-20833	New	New Feature [ADC] Add support for S32K118 from S32K14X platform Detailed description (how to reproduce it): Add support for S32k118, S32118 have only one ADC hardware Unit, we need update driver follow this Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: NA Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA
MCAL-20839	Bug	[LIN] Misra error<*>NewWorkDescription: <*> Please see the attachment file to get more detailed information <*> Requirement source: <*>NA <*> (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*> Please see the analysis tab
MCAL-20840	Bug	[SPI] Misra error<*>NewWorkDescription: Please see the attachment file to get more detailed information <*> Proposed solution (Optional): <*>* Update and fix misra for S32K14X_4.2 <*>+ In Spi.c. A part of Spi_WriteIB function.yP2CONST(Spi_DataBufferType, AUTOMATIC, SPI_APPL_CONST) pcDataBufferSrc=NULL_PTR * + InySpi_RegOperations.m file:y/* @violates @ref Spi[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!

Table continues on the next page...

ID	Subtype	Headline and Description
		ENDIF!]_PBcfg_c_REF_4 MISRA 2004 Required Rule 8.10, external linkage ...
MCAL-20841	New	New Feature [ISELED] Update driver according to API specification Rev 0.4, June 2018 NewWorkDescription: * N/A Requirement source: * Implementation Specification Document (ISP_INLC Rev. 0.4, June 2018) (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): * Create the new APIs introduced in the latest API specification.
MCAL-20842	New	New Feature [ISELED] Implement DMA support NewWorkDescription: * N/A Requirement source: * Customer Request (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): * Add DMA support.
MCAL-20844	Bug	[I2C] Change module name to I2c<*>Detailed description (how to reproduce it): <*> Arccoding to the requirement (General Requirements on Basic Software Modules AUTOSAR Release 4.2.2) - AUTOSAR_SRS_BSWGeneral.pdf <*> "[SRS_BSW_00300] All AUTOSAR Basic Software Modules shall be identified by an unambiguous name <*> Convention for module related files: <*>- < Module name>_*.* <*> - Spelling of module name: First letter of each word upper case, <*>consecutive letters lower case" <*> Please see the attachment file to get more information. <*> So, need to change naming from I2C to I2c. <*>1. I2C files <*>- I2C.mak <*>- I2C.xdm <*>- I2C_4.2.xdm <*>- I2C_Cfg.h <*>- I2C_Cfg_4.2.h 2. Function naming <*>3. Documents <*> Preconditions: <*>NA <*> Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*>Module naming is I2C <*>Expected behavior: <*>I2c <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>Rename
MCAL-20845	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[BASE] Change module name from I2C to I2c Detailed description (how to reproduce it): To synchronize the names of the AUTOSAR modules, Dev suggest changing the I2C name to I2c: 1. Currently: modules.h #define USE_I2C_MODULE ([!IF "node:exists(as:modconf("I2C")[1])"!]STD_ON[!ELSE!]STD_OFF[!ENDIF!]) 2. Change: #define USE_I2C_MODULE ([!IF "node:exists(as:modconf("I2c")[1])"!]STD_ON[!ELSE!]STD_OFF[!ENDIF!]) Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Module naming is I2C Expected behavior: I2c Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Rename
MCAL-20846	New	New Feature [RESOURCE] Change module name from I2C to I2c Detailed description (how to reproduce it): To synchronize the names of the AUTOSAR modules, Dev suggest changing the I2C name to I2c: 1. Currently: Resource.cfg RES_MODULES = Adc Can Dio Fls Gpt I2C Icu Lin Mcu Port Pwm Spi Wdg 2. Change: RES_MODULES = Adc Can Dio Fls Gpt I2c Icu Lin Mcu Port Pwm Spi Wdg Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Module naming is I2C Expected behavior: I2c Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Rename
MCAL-20847	New	New Feature [PWM] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-20903	Bug	[PWM] Fix misra violation<*>Detailed description (how to reproduce it):[...] Preconditions: <*>[...] <*> Test Case ID (internal TC that caught the defect) - optional <*>[...] Observed behavior: Expected behavior: Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional):
MCAL-20904	Bug	[ICU] [FTM] ICUFlexTimerAlternatePrescaler can not config when IcuEnableDualClockMode be enabled<*> Detailed description (how to reproduce it): User can not configyFlexTimer Alternate Prescaler althoughyIcuEnableDualClockModey was enabled Preconditions: Test Case ID (internal TC that caught the defect) - optional <*>[...] <*> Observed behavior: Expected behavior: Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): fix Xpath in icu.xdm <*><a:da name="EDITABLE" type="XPath"> <*><a:tst expr="../../../../IcuNonAUTOSAR/IcuEnableDualClockMode ='true'"/>
MCAL-20915	Bug	[I2C] Mismatch between the name of generated variant files and information within<*> Detailed description (how to reproduce it): <*> Reported Baseline: <*> The name of generated variant files is different from the one in the content of that file. <*>

Table continues on the next page...

ID	Subtype	Headline and Description
		For example, when generating MCU configuration with variant VS02, the generated file's name is:yMcu_VS02_PBcfg.c. However, in the content, it isyMcu_PBcfg_VS02.c instead. Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*> NA <*> Observed behavior: Mismatch naming of the generated variant file and the one in the file. <*> Expected behavior: <*> The naming should be aligned. <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*> Replace: I2c_PBcfg[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!] to: I2c[!IF "var:defined('postBuildVariant')"!]_[!"$postBuildVariant"!][!ENDIF!]_PBcfg
MCAL-20918	Bug	[MCU] READONLY attribute is evaluated by XPATH accessing resource file should be replaced by EDITABLE attribute<*> Detailed description (how to reproduce it): READONLY attribute with xpath have to be replace by EDITABLE with xpath. <*> Preconditions: <*>NA <*> Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*>NA <*> Expected behavior: <*>NA <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>NA
MCAL-20919	Bug	[MCU] Error in configuration when SOSC with maximum frequency<*> Detailed description (how to reproduce it): <*> McuSOSCFrequency configure with maximum value: 40000000 <*> with 118 platform: <*> Error because McuSPLLInputFrequency is READONLY and cannot be modify. <*> Preconditions: <*>NA <*> Test Case ID (internal TC that caught the defect) - optional <*>NA <*> Observed behavior: <*> McuSOSCFrequency configure with maximum value: 40000000 <*> ywith 118 platform: <*> Error because McuSPLLInputFrequency is READONLY and cannot be modify. <*> Expected behavior: <*> McuSOSCFrequency configure with maximum value: 40000000 <*>No Error McuSPLLInputFrequency <*> Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*> Proposed solution (Optional): <*>fix file mcu.xdm

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-20921	New	New Feature [DIO] Separate doc folder for each of Autosar version NewWorkDescription: In <Mdl>.mak file,yUM/IM is separated in doc, doc_4.2 and doc_4.3.yTherefore, UM/IM also need to be added in the following new folderyto be able to compile plugin: -ydoc_4.2/user_manuals -ydoc_4.3/user_manuals yRequirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add UM/IM in doc_4.2/user_manuals and doc_4.3/user_manuals folder.
MCAL-20922	New	New Feature [PORT] Separate doc folder for each of Autosar version NewWorkDescription: In <Mdl>.mak file,yUM/IM is separated in doc, doc_4.2 and doc_4.3.yTherefore, UM/IM also need to be added in the following new folderyto be able to compile plugin: -ydoc_4.2/user_manuals -ydoc_4.3/user_manuals yRequirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add UM/IM in doc_4.2/user_manuals and doc_4.3/user_manuals folder.
MCAL-20923	New	New Feature [ADC] Separate doc folder for each of Autosar version NewWorkDescription: In <Mdl>.mak file,yUM/IM is separated in doc, doc_4.2 and doc_4.3.yTherefore, UM/IM also need to be added in the following new folderyto be able to compile plugin: -ydoc_4.2/user_manuals -ydoc_4.3/user_manuals yRequirement source: NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add UM/IM in doc_4.2/user_manuals and doc_4.3/user_manuals folder.
MCAL-20942	New	New Feature [Mcem] fix tab in driver for S32K1xx y

Table continues on the next page...

ID	Subtype	Headline and Description
		* Detailed description (how to reproduce it): There are some tabulators reported in the latest driver Mcem_Eirm.cy Mcem_Eirm_Irq.cy Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): * fix tab
MCAL-20972	Bug	[FLS] Fixed the tabulators for Fls module<*>_*Detailed description (how to reproduce it):*_ <*> Some tabulatorsy exit in Fls_PBcfg_42.c file <*>_*Expected behavior:*_ <*> Remove allytabulatorsy in Fls_PBcfg_42.c file <*>_*Note:* in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)_ <*>_*Proposed solution -* optional:_ <*>[...]
MCAL-20974	New	New Feature [ISELED] Implement DMA burst mode _*NewWorkDescription:*_ * A new function called "digLED_Send_Cmd_Block" should be added to the current API so that multiple commands can be sent to the same ISELED strip in the shortest amount of time by making use of the DMA. _*Requirement source:*_ * Customer Request. (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) _*Proposed solution -* optional:_ * Implement the function specified in the description and correct any DMA related issue.
MCAL-20976	New	New Feature [ADC] Fix VSMD report for S32K14X_4.2 _*NewWorkDescription:*_ Error creating VSMD report. Please update: <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v mclass="PreCompile">VariantPostBuild</icc:v> <icc:v mclass="PreCompile">VariantPreCompile</icc:v> </a:a> For: AdcInterrupt in Adc_42.xdm. Please Fix:

Table continues on the next page...

ID	Subtype	Headline and Description
		<icc:v vclass="PostBuild">VariantPostBuild</icc:v> <icc:v vclass="PreCompile">VariantPreCompile</icc:v> To: <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> For: AdcChannelDelay in Adc_42.xdm. _*Requirement source:*_ [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) _*Proposed solution -* optional:_ [...]
MCAL-20979	Bug	[ADC] ADC failed to start group continous without interrupt<*>_*Detailed description (how to reproduce it):*_ <*>In Adc.xdm, continous without interrupt group can be configure when node "Adc Continous Without Interrupt Uses" is not enable, so in Adc_CfgDefines.h, ADC_CONTINUOUS_NO_INTERRUPT_SUPPORTED was not define. And in Adc_Adc12bsarv2.c, function Adc_Adc12bsarv2_StartNormalConversion, block code: <*>#ifdef ADC_CONTINUOUS_NO_INTERRUPT_SUPPORTED <*>/* If the continuous group is configured without interrupts then PDB will be configured as continuous mode */ <*>if((ADC_CONV_MODE_CONTINUOUS == pGroupPtr->eMode) && ((uint8)STD_ON == pGroupPtr->u8AdcWithoutInterrupt)) <*>{ <*>/* PDB operation in Continuous mode */ <*>Adc_Pdb_SetPdbMode(Unit, (boolean)TRUE); <*>} <*>else <*>{ <*>/* PDB operation in One-Shot mode */ <*>Adc_Pdb_SetPdbMode(Unit, (boolean)FALSE); <*>} <*>#endif /* ADC_CONTINUOUS_NO_INTERRUPT_SUPPORTED */ <*>was not compile, and group always start like One shot group <*>_*Preconditions:*_ <*>+ Adc Continuous Without Interrupt Uses is not enable <*>+ Adc group configure continous conversion mode, without interrupt is enable <*>_*Test Case ID (internal TC that caught the defect) -* optional:_ <*>NA <*>_*Observed behavior:*_ <*>NA <*>_*Expected behavior:*_ <*>NA <*>_*Note:* in the Expected behavior field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)_ <*>_*Proposed solution -* optional:_ <*>Configure group continous, without interrupt must be depend on NonAutosar node "AdcContinuousWithoutInterrupt"
MCAL-20984	New	New Feature [Base]Add memory and pointer classes for OCU in Compiler_Cfg.h _*NewWorkDescription:*_ InyCompiler_Cfg.h of Base driver Addymemory and pointer classes for module OCU #define OCU_CODE #define OCU_CONST #define OCU_APPL_DATA #define OCU_APPL_CONST #define OCU_APPL_CODE #define OCU_CALLOUT_CODE #define OCU_VAR_NOINIT #define OCU_VAR_POWER_ON_INIT #define OCU_VAR_FAST

Table continues on the next page...

ID	Subtype	Headline and Description
		#define OCU_VAR y _*Requirement source:*_ [...] (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) _*Proposed solution -* optional:_ [...]
MCAL-20990	New	New Feature [I2C] Rename generate files for multi-variants case _*NewWorkDescription:*_ Rename some generate files: I2c_Cfg.h I2c_Cfg_42.c I2c_Cfg.c I2c_PBcfg.c I2c_PBcfg_42.c I2c_Cfg_42.h I2c_RegOperations_42.m I2c_VersionCheck_Inc.m I2c_VersionCheck_Src.m I2c_VersionCheck_Src_PB.m to: CDD_I2c_Cfg.h CDD_I2c_Cfg_42.c CDD_I2c_Cfg.c CDD_I2c_PBcfg.c CDD_I2c_PBcfg_42.c CDD_I2c_Cfg_42.h CDD_I2c_RegOperations_42.m CDD_I2c_VersionCheck_Inc.m CDD_I2c_VersionCheck_Src.m CDD_I2c_VersionCheck_Src_PB.m and modified content in I2c.mak file _*Requirement source:*_ NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) _*Proposed solution -* optional:_ Please see the above text
MCAL-20992	New	New Feature [MCU] Fix misra for S32K14x RTM 1.0.1 ASR 4.2 _*NewWorkDescription:*_ comment misra error _*Requirement source:*_ NA (e.g.cPRD,gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) _*Proposed solution -* optional:_

4.3 RTM 1.0.0

ID	Subtype	Headline and Description
MCAL-7	New	New Feature [DEM] Move driver to Git and update according to ASR4.2.2 NewWork Description: Move driver to Git configuration management system. Update the folder structure and makefiles for drivers and development tests according to slides 5-10 from attached presentation. Use a stable version of the code for the migration - internal baselines. Both the source baselines and the tag applied in Git will be recorded in the Analysis tab, for further reference. Update the driver according to ASR4.2.2 specifications. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A" MCAL-8,New Feature,"[DET] Move driver to Git and update according to ASR4.2.2 NewWork Description: Move driver to Git configuration management system. Update the folder structure and makefiles for drivers and development tests according to slides 5-10 from attached presentation. Use a stable version of the code for the migration - internal baselines. Both the source baselines and the tag applied in Git will be recorded in the Analysis tab, for further reference. Update the driver according to ASR4.2.2 specifications. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A" MCAL-9,New Feature,"[ECUC] Move driver to Git and update according to ASR4.2.2 NewWork Description: Move driver to Git configuration management system. Update the folder structure and makefiles for drivers and development tests according to slides 5-10 from attached presentation. Use a stable version of the code for the migration - internal baselines. Both the source baselines and the tag applied in Git will be recorded in the Analysis tab, for further reference. Update the driver according to ASR4.2.2 specifications. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A"

Table continues on the next page...

ID	Subtype	Headline and Description
		MCAL-10,New Feature,"[ECUM] Move driver to Git and update according to ASR4.2.2 NewWork Description: Move driver to Git configuration management system. Update the folder structure and makefiles for drivers and development tests according to slides 5-10 from attached presentation. Use a stable version of the code for the migration - internal baselines. Both the source baselines and the tag applied in Git will be recorded in the Analysis tab, for further reference. Update the driver according to ASR4.2.2 specifications. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A" MCAL-12,New Feature,"[RTE] Move driver to Git and update according to ASR4.2.2 NewWork Description: Move driver to Git configuration management system. Update the folder structure and makefiles for drivers and development tests according to slides 5-10 from attached presentation. Use a stable version of the code for the migration - internal baselines. Both the source baselines and the tag applied in Git will be recorded in the Analysis tab, for further reference. Update the driver according to ASR4.2.2 specifications. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-11705	Bug	[GPT] Fix Predefined Timers findings when ussing FTM or LPIT or LPTMR<*>Problem detailed description (how to reproduce it):<*>The RANGE atribute on predef timers should take in account configuring the frequency in MCU<*>(it is possible to have 1Mhz frequency for LPIT from MCU for example).<*>The default value for FTM prescaler should take in account hardware specifics.<*>- The correct frequency is FTM_Initial / (2 ^ presc).<*>- Freq for LPTMR is not checked and correctly set<*>Preconditions:<*>The user uses predefined timers.<*>Observed behavior:<*>Some issues in generating predefined timers for FTM and LPIT<*>Expected behavior:<*>Correct use of predefined timers.<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>[...]
MCAL-13926	Bug	[PWM] Incorrect Compiler Abstraction keyword<*>Problem detailed description (how to reproduce it): <*>This declaration is not follow coding convention and does not have Compiler Abstraction keyword <*>void Pwm_Ftm_SelectCommonTimebase <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>This declaration is not follow coding convention and

Table continues on the next page...

ID	Subtype	Headline and Description
		does not have Compiler Abstraction keyword <*>void Pwm_Ftm_SelectCommonTimebase <*>Expected behavior: <*>This declaration need to follow coding convention and have Compiler Abstraction keyword <*>void Pwm_Ftm_SelectCommonTimebase <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>The declaration should to be like this: <*>FUNC(void, PWM_CODE) Pwm_Ftm_SelectCommonTimebase
MCAL-14348	New	New Feature [DET] Add support for DET Runtime error reporting NewWork Description: AUTOSAR 4.2.1 introduced two new APIs in DET specification: Det_ReportRountineError and Det_ReportTransientFault. As of AUTOSAR 4.2.2, the FLS specification requires reporting runtime errors to Det_ReportRountineError(), thus needing support in the DET MCAL stub. Requirement source: AUTOSAR 4.2.2 DET(SWS_Det_01001) and FLS SWS document Proposed solution (Optional): - Add Det_TestLastReportRuntimeError() API in build env Det_stub.c and Det_stub.h - Add Det_ReportRuntimeError() API in Det.c and Det.h driver files.
MCAL-14391	New	New Feature [ADC] Remove some unused defines in S32K14X Problem detailed description (how to reproduce it): There are some defines generated in Adc driver that are not used anywhere, so they should be removed: ADC_STREAMING_NUM_SAMPLES ADC_SKIP_DMA_CONTINUOUS_CHECK ADC_MAX_CHANNEL_PER_HW_UNIT ADC_MAX_GROUPS_PER_HW_UNIT ADC_MAXIMUM_HW_CHANNELS Preconditions: Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: N/A Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove the unused definitions
MCAL-14393	New	New Feature [ADC] Improve the Adc_Adc12bsarv2_StartDmaOperation function

Table continues on the next page...

ID	Subtype	Headline and Description

		NewWork Description:
		In the Adc_Adc12bsarv2_StartDmaOperation function, the condition to check hardware unit enabled or not should be perfomred only once . See the lines of code below:
		if ((uint8)STD_ON == Adc_pCfgPtr->Misc.au8Adc_HwUnit[Unit])
		{
		/* Disable DMA interrupt */
		Mcl_DmaDisableHwRequest(DmaChannel);
		#if (MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON)
		Mcl_DmaDisableNotification(DmaChannel);
		#endif
		}
		/** @violates @ref Adc_Adc12bsarv2_c_REF_8 Array indexing shall be the only allowed form of pointer arithmetic */
		pResult = pGroupPtr->pResultsBufferPtr[Group] + Adc_aGroupStatus[Group].ResultIndex;
		if ((uint8)STD_ON == Adc_pCfgPtr->Misc.au8Adc_HwUnit[Unit])
		{
		#if (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING)
		if ((boolean) TRUE == pGroupPtr->bAdcDoubleBuffering)
		{
		...
		It should be better when merge them into one condition like this:
		if ((uint8)STD_ON == Adc_pCfgPtr->Misc.au8Adc_HwUnit[Unit])
		{
		DmaChannel = (Mcl_ChannelType)Adc_pCfgPtr->Misc.au8Adc_DmaChannel[Unit];
		/* Disable DMA interrupt */
		Mcl_DmaDisableHwRequest(DmaChannel);
		#if (MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON)
		Mcl_DmaDisableNotification(DmaChannel);
		#endif
		/** @violates @ref Adc_Adc12bsarv2_c_REF_8 Array indexing shall be the only allowed form of pointer arithmetic */
		pResult = pGroupPtr->pResultsBufferPtr[Group] + Adc_aGroupStatus[Group].ResultIndex;
		#if (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING)
		if ((boolean) TRUE == pGroupPtr->bAdcDoubleBuffering)
		{
		In addition: returnValue variable in Adc_Adc12bsarv2_SetClockMode function should be remove and the return should be E_OK.
		Requirement source:
		N/A
		(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
		Proposed solution (Optional):
		Update Adc_Adc12bsarv2_StartDmaOperation function as the description.

MCAL-14394	New	New Feature [ADC] Update the description for AdcTimeout element in xdm file NewWork Description: The description for AdcTimeout element is not correct for S32K14X: This is a timeout value which is used to wait till - the conversion is not aborted - ADC hardware is not entered in power down state

Table continues on the next page...

ID	Subtype	Headline and Description
		- ADC hardware is not entered in idle state If the Status is not updated then after this timeout the ADC_E_TIMEOUT production error will be reported and the rest of the functionality will be skipped Because S32K14X does not have power down state, It should be updated. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-14444	Bug	[ADC] Incomplete implementation of CPR-MCAL-791.adc<*>Problem detailed description (how to reproduce it):<*>If AdcEnableDoubleBuffering is used for a HW triggered group configured with circular buffer, only the first stream of conversions will be executed.<*>Preconditions:<*>AdcEnableDoubleBuffering is confiugred for a streming group with circular buffer<*>Test Case ID (internal TC that caught the defect) - optional<*>n/a<*>Trigger:<*>n/a<*>Observed behavior:<*>Streaming group does not work<*>Expected behavior:<*>Continuous streams of conversions will be executed, not just the first one.<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>Remove DMA_TCD_DISABLE_REQ_U8 flag form DMA confiugration in case ADC_ENABLE_DOUBLE_BUFFERING is enabled and the group uses double buffering feature.
MCAL-14486	New	New Feature [MCL] Update priority check for Mcl Cross bar Priority Master NewWork Description: Some platforms have master ID not continuous: Master0,Master1,Master5,Master6. Update the MclCrossbarPrioMaster implementation to support this use case Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Define max value priority of master in resource file. Implement new range check.
MCAL-14562	New	New Feature [WDG] Investigate and improve cyclomatic complexity and nesting level NewWork Description: In WDG driver ASR ver 4.2.x. The function Wdg_ChannelValidateMode has cyclomatic complexity value greater 10. Requirement source: - CPR-MCAL-783: The cyclomatic complexity of each module shall be in the range of 0 to 20. A warning shall be generated for cyclomatic complexity values between [10...20]. An error shall be generated for cyclomatic complexity values greater than 20.

Table continues on the next page...

ID	Subtype	Headline and Description
		The requirements can be marked as Fulfilled In in the module has the maximum cyclomatic complexity lower than 20 and all the warnings are commented out. - CPR-MCAL-784: The nesting level of conditionals in each module shall be in the range of 0 to 4. An error shall be generated for a nesting level greater than 4. The requirements can be marked as Fulfilled In in the module has the maximum nesting level lower than 4 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update source code to make lower its cyclomatic complexity value
MCAL-14674	Bug	[ADC] The variable Adc_aRuntimeGroupChannel[Group].bRuntimeUpdated is not initialized before being used<*>Problem detailed description (how to reproduce it): <*>The variable Adc_aRuntimeGroupChannel[Group].bRuntimeUpdated should be initialized with 0 before being used in any functions. <*>The function Adc_InitGroupsStatus should have initialized the variable. In the case of value of the variable is 1 the program might crash by the following sequence: <*>- Adc_Init() <*>- Adc_StartGroupConversion(); /* NULL pointer will be used this function */ <*>If ADC driver erroneously sees that Adc_setChannel was called by checking the flag, bRuntimeUpdated, it will try to access the new Group configuration stored in Adc_aRuntimeGroupChannel[Group].pChannel variable. <*>But if Adc_SetChannel wasn't actually called, the pointer could be NULL or point to some invalid memory. <*>However, Adc_aRuntimeGroupChannel should be placed in .bss section and initialized with zeros by the startup code, in which case the issue does not appear. <*>Preconditions: <*>Adc_SetChannel is enabled. <*>Adc_aRuntimeGroupChannel structure is not zero-ed by startup code (it should be in .bss section) <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Trigger: <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A
MCAL-14675	Bug	[ADC] The channel configuration of a group will not be updated by the function Adc_SetChannel after stopping and restarting a group<*>Problem detailed description (how to reproduce it): <*>The channel configuration of a group will not be updated by the function Adc_SetChannel if the group is stopped and then restarted. <*>API sequence to reproduce the issue: <*>Adc_Init(); <*>Adc_StartGroupConversion(G0); <*>Adc_SetChannel(G0,...) <*>- wait till the group finish conversion and channels are updated <*>Adc_StopGroupConversion(G0,); <*>- Adc_StartGroupConversion(G0) <*>-> Now channel list of G0 is not update instead of primary channels list. <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Trigger: <*>N/A <*>Observed behavior: <*>The original configuration of the group is used instead of the new configuration that was updated via Adc_SetChannel. implementation of CPR-MCAL-797.adc is incomplete. <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-14705	Bug	[GPT] Correct exclusive areas to protect eChannelStatus<*>Problem detailed description (how to reproduce it):<*>- Gpt_StartTimer: Gpt_aChannelInfo[channel].eChannelStatus and STM/PIT/TOM channel enabled (TOM counter clear) should be placed in the same exclusive area. As the comment in Gpt_StartTimer, the Gpt_aChannelInfo[channel].eChannelStatus should be updated before enable HW channel.<*>- Gpt_StopTimer: eChannelStatus test and update should be placed in exclusive area. The HW channel should be stopped before updating eChannelStatus.<*>- Gpt_EnableNotification/Gpt_DisableNotification: eChannelStatus test and update should be placed in exclusive area<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Trigger:<*>NA<*>Observed behavior:<*>Some cases eChannelStatus get unexpected value<*>Expected behavior:<*>eChannelStatus is fully protected<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>NA
MCAL-14760	New	New Feature [BASE] Migrate specific files for S32K from CC to GIT Migrate the Base specific files for S32K from CC to GIT
MCAL-14761	New	New Feature [RESOURCE] Migrate S32K specific files from CC to GIT Migrate S32K specific files from CC to GIT
MCAL-14817	New	New Feature [CAN] Improve function for update transmit data in FD and non-FD NewWork Description: There are 2 functions to update transmit data for FD and non-FD. These 2 functions share a lot of similarities. It would be better to merge together with consideration to cyclomatic level. Additionally, it is better to move the check for length to new function for used in different place. This also reduce the cyclomatic level of current function. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA
MCAL-14828	New	New Feature [MCU] Update extra information for ClkReferencePoints NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Update extra information for ClkReferencePoints in MCU document. Clock reference points must be checked manually to be the same as the hardware selected clock in Kinetis. Unfortunately, this information is not written in any document released making user confused about ClkReferencePoints class. No information for restrain between McuClkReferencePoints and McuPeripheralClkConfig is available in the manual. Expected behavior: The description for McuClkReferencePoints should be clearer Requirement source: Internal task (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update in the xdm the description of McuClkReferencePoints and also in the user manual. The information should clearly state that the clock frequency configured in McuPeripheralClkConfig should be used to export the clock frequency through McuClkReferencePoints. This reference point should be used in the configuration of the module that uses it (SPI, I2C, GPT, etc.). If the configured module has also an internal clock selection (like FlexTimer for example) the clock reference point should be configured taking the internal clock selection into account and the reference used should reflect the clock that finally enters the used peripheral.
MCAL-14833	New	New Feature [CAN] The CanClockFromBus parameter should be gray out or removed when CTRL[CLK_SRC] is not used. Problem detailed description (how to reproduce it): The customer reported that they was confused by the parameter CanClockFromBus "Can Clock from Bus (CTRL[CLK_SRC])" even this bit is not used in S32V234 as stated in RM : "CAN_CTRL1[CLKSRC] bit is not used. For selecting PE clock between FXOSC and PLL, configure CAN_CLK(MC_CGM_0 - AUX6_DIV0) accordingly. Please refer clocking chapter more information" Could you gray it out with checked or remove this parameter from S32V234? Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update configuration to generate correctly config which the platforms support CAN_CTRL1[CLKSRC] bit.
MCAL-14838	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[CAN] Improve implementation of separating message buffer size () New work description: There are some improvements required for Can driver after introducing feature allow user to configure different message buffer size in different RAM region. - Pointer pCanMbConfigContainer did not declared in function Can_FlexCan_Cancel() in Can_Flexcan.c. - In Can_PBcfg.c: Has incorrect character (this is '[') before CAN_CONTROLLERCONFIG_OVER_EN_U32. - Variable u16MBMapping[] is not used when interrupt is disabled. Furthermore, should not use u16MBMapping[] to determine u32Hrh in function Can_FlexCan_RxFifoFrameAvNotif() , because the value of element whose index is (Can_pControlerDescriptors[controller].u8RxFiFoUsedMb-1U) in array u16MBMapping[] is invalid. Driver should determine u32Hrh based on the array which stored information of Hardware objects. - Don't need to check enabled Rxfifo in function Can_FlexCan_ProcessRxFifo because it was checked before calling to the function. - The local functions should be static function, including: Can_FlexCan_ProcessTxPoll, Can_FlexCan_ProcessRxFifo, Can_FlexCan_ProcessRxNormal - Remove variables which do not use, included: Can_u8RealPayloadData, - Fix some compiler warnings - Should have a variable to record the first index of HTH, then using the variable for process RX and TX - Controller will incorrectly cancel MB which has lower priority, cause it was incorrect input parameter for Can_FlexCan_Cancel() function in Can_FlexCan_Write(). This issue only impact to ASR 4.0.
MCAL-14842	New	New Feature [CAN] Move Can_SourceClock.m file from specific folder to generic folder Problem detailed description (how to reproduce it): Can_SourceClock.m is used to define the macro which get the source clock for FlexCan module from MCU module. This file is same for all platform, so it should be located in generic folder than in specific folder. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): - Move Can_SourceClock.m from can\specific\<Platform_Name>\generate to can\generic\generate\flexcan and can\generic\generate\flexcan_42.
MCAL-14845	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[CAN] When the CAN FD feature is enabled, always use the CAN_CBT register for the CAN bit timing NewWork Description: According to ticket MCAL-80 (Revire new Reference manual for Calypso 3M/6M), there are some changes which are related to CAN_CBT register need to be resolve, including: 1. Note in (43.4.3): When the CAN FD feature is enabled, do not use the PRESDIV, RJW, PSEG1, PSEG2, and PROPSEG fields of the CAN_CTRL1 register for CAN bit timing. Instead use the the CAN_CBT register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields. + Furthermore, Note in 43.5.9.7: When the CAN FD feature is enabled, always set CAN_CBT[BTF] and configure the CAN bit timing variables in CAN_CBT. See CAN Bit Timing Register (CAN_CBT). 2. Change some field bit as followed: + The ERJW field in CAN_CBT register is extended from 4 bits up to 5 bits (see 43.4.19). 3. "If the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase and will last until the next synchronization event. Thus, the length of the time quantum should be the same in nominal and data bit timing in order to minimize the chance of error frames on the CAN bus, and to optimize the clock tolerance in networks that use FD frames." (see 43.5.9.2 CAN FD frames)
MCAL-14847	New	New Feature [CAN] Add recommendation timeout value and improve freeze mode entried In latest version of RM for Calypso (rev 5 RC), there is a section for entrying freeze mode: On this chip, the procedure to enter Freeze mode is: 1. Set both CAN_MCR[FRZ] (Freeze Enable) and CAN_MCR[HALT] (Halt) to 1. 2. Check whether CAN_MCR[MDIS] (Module Disable) is set to 1. If it is, clear it to 0. 3. Poll the MCR register until CAN_MCR[FRZACK] (Freeze Mode Acknowledge) is set to 1 or the timeout is reached. NOTE The minimum timeout duration must be equivalent to: a. 730 CAN Nominal bits if CAN_MCR[FDEN] (CAN FD Operation Enable) is set to 1 (CAN bits calculated at arbitration bit rate), b. 180 CAN bits if CAN_MCR[FDEN] is cleared to 0. 4. If CAN_MCR[FRZACK] is set to 1, no further action is required. Skip steps 5to 8. 5. If the timeout is reached because CAN_MCR[FRZACK] remains cleared to 0, then set CAN_MCR[SOFTRST] (Soft Reset) to 1. 6. Poll MCR until CAN_MCR[SOFTRST] is cleared to 0. 7. Reconfigure the Module Control Register (CAN_MCR). 8. Reconfigure all the Interrupt Mask Registers (CAN_IMASKn). After these steps are complete, the module is in Freeze mode. This provides recommendation value for configuring timeout. Additionally, the errors case (timeout occured) similar to workaround defined by errata ID ERR_IPV_FLEXCAN_0014 (e8341, e9595 - for Cobra55). Therefore, it would

Table continues on the next page...

ID	Subtype	Headline and Description
		be better to make the workaround as normal behavior of the driver.
MCAL-14855	Bug	[MCL] Naming of MclDemEventParameterRefs is not ASR conform<*>Problem detailed description (how to reproduce it): <*>In ASR 4.2 AUTOSAR_SWS_BSWGeneral.pdf: <*>[SWS_BSW_00125] Naming convention for Error values <*>Error values shall be named in the following way: <*><MIP>_E_<EN> <*>Where here <MIP> is the Capitalized module implementation prefix of this BSW <*>Module (SWS_BSW_00102) <EN> is the error name. Only capital letters shall be <*>used. If <EN> consists of several words, they shall be separated by <*>underscore. (SRS_BSW_00327) <*>Accordingly, it should be MCL_E_<EN> instead of MCL_DMA_E_<EN> as in current release version. <*>In ASR 4.0.3 the same naming convention is stated by BSW00327. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>Customer <*>Observed behavior: <*>Incorrect naming of errors. <*>Expected behavior: <*>Move "_DMA" into <EN> part, behind "_E_". <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>MCL_E_DMA_DESCRIPTOR <*>MCL_E_DMA_ECC <*>MCL_E_DMA_BUS
MCAL-14870	Bug	[CAN] Incorrectly determine the index of MB (MB0) which is assigned for RxFifo<*>NewWork Description: <*>- Incorrectly determine the index of MB (MB0) which is assigned for RxFifo. <*>/* Find the global index of the first MB (MB0) of this controller.*/ <*>while ((u16MBGlobalIndex < (uint16)Can_pCurrentConfig->uCanFirstHTHIndex) && <*>(controller == (((pCanMbConfigContainer->pMessageBufferConfigsPtr) + u16MBGlobalIndex)->u8ControllerId)) && <*>((uint8)0U != (((pCanMbConfigContainer->pMessageBufferConfigsPtr) + u16MBGlobalIndex)->u8HWMBIndex)) <*>) <*>{ <*>u16MBGlobalIndex ++; <*>} <*>- The reason of this issue is incorrect condition in while statement. Correct it as followed <*>while ((u16MBGlobalIndex < (uint16)Can_pCurrentConfig->uCanFirstHTHIndex) && <*>((controller != (((pCanMbConfigContainer->pMessageBufferConfigsPtr) + u16MBGlobalIndex)->u8ControllerId)) || <*>((uint8)0U != (((pCanMbConfigContainer->pMessageBufferConfigsPtr) + u16MBGlobalIndex)->u8HWMBIndex))) <*>) <*>{ <*>u16MBGlobalIndex ++; <*>} <*>
MCAL-14878	Bug	[CAN] Incorrectly determine the controller ID of MBs in multi transmission mode<*>Problem Description: <*>+ In function Can_FlexCan_MainFunctionWrite and Can_FlexCan_MainFunctionMultipleWritePoll, it need to parse through all TX configured message buffer of the init controller. <*>The driver incorrectly determines controller ID of HTH in case multi transmission mode is enabled because the controller ID is determined based on u8ControllerIdMapping array, but the number of element of this array is less than the number of configured HOHs. <*>Purpose solution: <*>+ Should use information from structure which record information of HOHs to determine controller ID of the HOHs. This statement is as following: <*>Can_pCurrentConfig->MBConfigContainer.pMessageBufferConfigsPtr[u16CtrlGlobalIndex].u8ControllerId <*>Where, u16CtrlGlobalIndex is the index of HOH which is parsed in the structure array which record information of HOHs <*>

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-14895	Bug	[CANIF] Macro CAN_PRETENDED_NETWORKING is not used<*>Problem Description: <*>CanIf driver uses macro CAN_PRETENDED_NETWORKING as a pre-compile tag to generated function CanIf_CurrentIcomConfiguration. This macro is defined in Can driver. However, this macro is going to be replaced from Can driver by CAN_PUBLIC_ICOM_SUPPORT in ticket MCAL-10340. Therefore, it is required to update CanIf driver to compatible with the latest Can Driver <*>Purpose solution: <*>- Replace CAN_PRETENDED_NETWORKING with CAN_PUBLIC_ICOM_SUPPORT
MCAL-14900	Bug	[CAN] Rx FIFO Overflow is not processed <*>Bug Description: <*>- When Rx FIFO Overflow occurs then BUF7I in CAN_IFLAG1 is set, but the driver is not process it because Can_FlexCan_ProcessRxFifo() function don't reach to bit BUF7I. <*>The problem happens with below statement: <*>for (u8MbIndex = (uint8)FLEXCAN_FIFOFRAME_INT_INDEX_U8; u8MbIndex < (uint8)FLEXCAN_FIFOOVERFLOW_INT_INDEX_U8; u8MbIndex++) <*>in this statement, u8MbIndex can not reach to FLEXCAN_FIFOOVERFLOW_INT_INDEX_U8 <*>Purpose solution: <*>- Modify the FOR statement as follwed: <*>for (u8MbIndex = (uint8)FLEXCAN_FIFOFRAME_INT_INDEX_U8; u8MbIndex <= (uint8)FLEXCAN_FIFOOVERFLOW_INT_INDEX_U8; u8MbIndex++) <*>- Do it in Can_FlexCan_MainFunctionRead and Can_FlexCan_MainFunctionMultipleReadPoll <*>
MCAL-14902	Bug	[MCL] The EMI Tresos field is editable but it has no impact on the code<*>The register DMA_DCHMID(containing the EMI field) is not present in the S32K14X platform, <*>so the Tresos EMI field must not be editable. <*>The issue is only cosmetic, it has no impact to the code. <*>
MCAL-14904	New	New Feature [MCL] Update the Tresos name for parameters ECP, EMI, DPA Update the Tresos name for parameters ECP, EMI, DPA: ECP- change to "Enable Channel Preemption" EMI - change to "Enable Master ID Replication" DPA - change to "Disable Preempt Ability" Proposed sollution: update the Mcl.xdm label attribute for the parameters ECP, EMI, DPA.
MCAL-14928	Bug	[CAN] In dual clock mode, Can driver is not reconfigured<*>Problem detailed description (how to reproduce it): <*>In the current driver, when using dual clock mode, driver only reconfigure the baud rate register and do not re-configure other registers. Therefore, the following issues could occurred: <*>- If controller is reset due to issue when switch to Stop mode, all registers are not configured with proper value. <*>- If there are pending received messages, it may incorrectly report to upper layer. <*>=> The driver should reconfigure all

Table continues on the next page...

ID	Subtype	Headline and Description
		registers when we change the Can mode to CAN_T_START. <*>Preconditions: <*>- The dual clock mode is enabled and the alternate clock mode is used <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>Can not receive the message from bus because the MB memories are not configured. <*>Expected behavior: <*>The driver should reconfigure all registers and memory area <*>Proposed solution (Optional): <*>- Can_FlexCan_SetClockMode changes value for a variable which determines either "normal clock" or "alternate clock". <*>- Always call Can_Flexcan_ChangeBaudrate when the driver sets the controller to CAN_T_START mode <*>- In Can_Flexcan_ChangeBaudrate performs to configure "normal clock" or "alternate clock" based on checking the value of the variable. <*>
MCAL-14929	Bug	[DIO] Fix the code after review against checklist<*>Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2
MCAL-14930	Bug	[PORT] Fix the code after review against checklist<*>Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2
MCAL-14952	Bug	[CAN] MAXMB value must take into account the region of MBs occupied by RxFIFO and its ID filters table space <*>Problem detailed description (how to reproduce it): <*>- When RxFifo is enabled and the only one HOH is assigned for the controller, the driver is generating zero for the MAXMB value. In the fact, according to Reference manual, the definition of MAXMB value must take into account the region of MBs occupied by RxFIFO and its ID filters table space defined by RFFN bit in CAN_CTRL2 register. <*>Preconditions: <*>- Rxfifo is enabled and the only one HOH is assigned for the controller <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>- When RxFifo is enabled and the only one HOH is assigned for the controller, the driver is generating zero for the MAXMB value <*>- The driver can not receive the messages with the different IDs <*>Expected behavior: <*>- MAXMB value must take into account the region of MBs occupied by RxFIFO and its ID filters table space defined by RFFN bit in CAN_CTRL2 register. <*>- The driver can receive the messages with the different IDs <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): NA <*>
MCAL-14958	Bug	[SPI] Fix misra report<*>Problem detailed description (how to reproduce it): <*>File "...\plugins\Spi_TS_T2D30M20I0R0\src\Spi.c", line 360, MISRA 2004 Rule Violated 8.7 (Required): no MISRA violation comment was found (maybe wrong format is used). <*>Line static volatile VAR(uint32, SPI_VAR) Spi_u32SpiBusySyncHWUnitsStatus = 0u;
MCAL-15001	Bug	[ADC] Correct the behavior of Adc_SetClockMode depending on

Table continues on the next page...

ID	Subtype	Headline and Description
		AdcConvTimeOnce<*>Problem detailed description (how to reproduce it):<*>The Adc_SetClockMode is not depend on Adc_SetMode and AdcConvTimeOnce anymore. I see a problem that we have alternate timing configuration for hardware unit in case alternate clock is used but don?t have this one for group. So in case AdcConvTimeOnce is disabled and alternate clock is used, the timing configuration for normal clock will be used. This is incorrect.<*>The problem is that if AdcConvTimeOnce is disabled, ADC should use the timing configuration from the groups. But currently the groups contain only settings for normal mode (no settings for alternate clock). Parameters for Unit (when AdcConvTimeOnce is enabled) already contain settings from normal and alternate clock.<*>So there is a need to add settings for "alternate clock" in the group settings.<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Trigger:<*>NA<*>Observed behavior:<*>NA<*>Expected behavior:<*>NA<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>To implement this, we need to change the xdm, adjust the group configuration, but also to store in some global variable the current clock mode and check it whenever a group conversion starts, and based on it - use the normal or alternate group timings.<*>If AdcConvTimeOnce is disabled and Adc_SetClockMode is called, I think the functions should not update any conversion timings, only the variable for clock mode will be updated. ; AdcAlternateConvTimings actually makes sense only if AdcConvTimeOnce is enabled. Otherwise, the group alternate conversion times will be used.<*>So Adc_SetClockMode does not depend on AdcConvTimeOnce, but it will have a different behavior depending on whether it?s enabled or not:<*>- If it?s enabled: configure the registers based on unit settings<*>- If it?s disabled: update only a global variable, then start group Apis will adjust the settings.
MCAL-15009	New	New Feature [GPT] Protect uninterruptible code Protected code which is not interruptable should be implemented as defined by {code:java} CPR-MCAL-822 {code} Gpt_Stm_StartTimer : * read the counter value * update the next compare value base on the counter value, plus timeout value The problem is : - after point1 ,a system interrupt occur, - the updated value in point2 too closed to point 1 -> the match event may not occur at current period of counter Solution: Use SuspendAllInterrupts() and ResumeAllInterrupts to protect above 2 steps
MCAL-15014	New	New Feature [CAN] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport.

Table continues on the next page...

ID	Subtype	Headline and Description
		The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		[?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15018	New	New Feature [FEE] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \

Table continues on the next page...

ID	Subtype	Headline and Description
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15029	Bug	[SPI] Implement findings from code review against checklist<*>NewWork Description: <*>- Review code against checklist. <*>- Fill the review result to the checklist. <*>- The checklist template and coding guideline are enclosed in attachment. <*>Requirement source: <*>sMCAL Release criteria document version 5.1: <*>http://compass.freescale.net/go/228798570 <*>Coding guideline verion 5.0 date 19-Jul-2016: https://www.nxp.com/go/230979668 <*>Code review checklist verion 4.0, date Nov-2016: https://www.nxp.com/go/230979668 <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>- The result of this activity will be the code review checklist and this ticket is treated as ?platform specific?. <*>- Code Review checklists, both Intermediate and Final, should be added in GIT and attached to this ticket. <*>- New ticket will be raised for the code modifications resulted for the ?code review? activity. This ticket will be used to update code, will be analyzed for all platforms and changes will be integrated for all affected platforms. <*>
MCAL-15032	Bug	[LIN] Correct misra errors for Calypso RTM 1.0.0<*>There are several misra errors in Lin_Cfg.c and Lin_PBcfg.c. <*>Please see the attached file for details.
MCAL-15034	Bug	[I2C] Uninterruptible sequences should be protected<*>Problem detailed description (how to reproduce it): <*>When using DMA for transfering data using an I2C FlexIO channel any interruptions between seding the slave address and enabling the DMA hardware requests will cause the transmission to fail. <*>Preconditions: <*>Using a FlexIO channel configured for DMA transfers. <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>Interruption of the code between sending the address and enabling the DMA requests <*>Observed behavior: <*>I2C transmission fails <*>Expected behavior: <*>I2C data is sent <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Guard the sequence with suspend and resume interrupts.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15035	New	New Feature [I2C] Update UM/IM for I2C to support both ASR 4.0 and 4.2 in Git Update UM/IM for I2C to support both ASR 4.0 and 4.2 in Git
MCAL-15038	Bug	[CAN] Not generate code for PNET for not supported platform<*>Bug description: <*>The code generator performs to generate the code for the PNET feature. And it only check from user's configuration (*xdm file in ASR 42), but it do not from the resource files, whether the PNET feature is used or not. The current code generator as following: <*>[!IF "CanGeneral/CanPublicIcomSupport = 'true'"!] <*>................ <*>[!ENDIF!] <*>However, the user may set or import an incorrect configuration and this will lead to an error that although the platform do not support PNET, the code for it generate. <*>Purpose solution: <*>- The code generator must check both two conditions from user's configuration and from the resource files before generating the code for PNET, as following: <*>[!IF "(CanGeneral/CanPublicIcomSupport = 'true') and (ecu:get('Can.CanConfigSet.CanPretendedNetworking')='STD_ON')"!] <*>........................... <*>[!ENDIF!] <*>
MCAL-15042	Bug	[SPI] Some tags only support PreCompile in Spi.xdm<*>Problem detailed description (how to reproduce it): <*>Have some tags only support PreCompile in Spi.xdm file: SpiCPUClockRef, SpiPhyUnitAlternateClockRef. <*>Because they will use for predefine parameters: SPI_TIMEOUT_COUNTER, SPI_DUAL_CLOCK_MODE. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>SpiCPUClockRef,SpiPhyUnitAlternateClockRef support for post-build selectable variants. <*>Expected behavior: <*>SpiCPUClockRef,SpiPhyUnitAlternateClockRef only support for PreCompile. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>- Remove <a:a name="POSTBUILDVARIANTVALUE" value="true"/> <*>- Replace: <*><icc:v vclass="Link">VariantLinkTime</icc:v> <*><icc:v vclass="PreCompile">VariantPreCompile</icc:v> <*><icc:v vclass="PostBuild">VariantPostBuild</icc:v> <*>With: <*><icc:v vclass="PreCompile">VariantLinkTime</icc:v> <*><icc:v vclass="PreCompile">VariantPreCompile</icc:v> <*><icc:v vclass="PreCompile">VariantPostBuild</icc:v>
MCAL-15043	Bug	[ADC] Adc_GetCMR_Register function should not be available for some platforms<*>Problem detailed description (how to reproduce it): <*>Adc_GetCMR_Register function does not used for S32K14X, MAC58RXXX, MPC5777C. So this function should not be available for these platforms to avoid getting failure at build. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>Adc_GetCMR_Register should not appear in plugin. <*>Expected behavior: <*>Test can be built successfully <*>Note: in the ?Expected behavior? field, please mention also the requirement source too

Table continues on the next page...

ID	Subtype	Headline and Description
		(cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Adding the condition for the appearance of Adc_GetCMR_Register
MCAL-15047	New	New Feature [WDG] Add support for running from User Mode on Treerunner and S32K NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else
		#define Call_Can_FlexCan_ResetController(Controller) \\

Table continues on the next page...

ID	Subtype	Headline and Description
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15050	New	New Feature [FLS] Add support for running from User Mode on S32K NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example:

Table continues on the next page...

ID	Subtype	Headline and Description
		You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15051	New	New Feature [RTE] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT

Table continues on the next page...

ID	Subtype	Headline and Description
		protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15052	New	New Feature [RESOURCE] Add CRCU in the list of resources for S32K14X platform NewWork Description: Add CRCU in the list of resources for S32K14X platform Requirement source: Customer request

Table continues on the next page...

ID	Subtype	Headline and Description
		(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add CRCU in the list of resources for S32K14X platform
MCAL-15060	Bug	[PWM] Update checking NULL pointer Pwm_Init<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>When PWM_PRECOMPILE_SUPPORT = STD_ON, the Pwm_ValidateParamPtrInit is still check pointer passed to Pwm_Init need difference NULL. DET error is always raise and driver cannot initial <*>Preconditions: <*>PWM_PRECOMPILE_SUPPORT = STD_ON <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>DET error is always raised and driver cannot initial <*>Expected behavior: <*>DET is not raised <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>NA <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>add c <*>hecking #if (PWM_PRECOMPILE_SUPPORT == STD_OFF)
MCAL-15064	Bug	[WDGIF] Incorrect file include structure of Watchdog Interface<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>According to WDGIF002, file include structure is wrongly implemented in MCAL driver. As described in Specification of Watchdog Interface, Wdg<xxx> shall be included in WdgIf.c instead of WdgIf.h due to the existence of WdgIf.c. <*>Release version: MPC574XG_MCAL4_0_RTM_1_0_2 <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the bug) - optional <*>[...] <*>Trigger (not applicable in case of new features): <*>[...] <*>Observed behavior (not applicable in case of new features): <*>State / Status: <*>Business Unit: <*>Project Area: <*>Parent Bin: <*>Subtype: <*>Severity: <*>Priority: <*>Headline: <*>Problem detailed description (how to reproduce it): <*>According to WDGIF002, file include structure is wrongly implemented in MCAL driver. As described in Specification of Watchdog Interface, Wdg<xxx> shall be included in WdgIf.c instead of WdgIf.h due to the existence of WdgIf.c. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>Wdg<xxx>.h is now included in WdgIf.h <*>Expected behavior: <*>Wdg<xxx>.h should be included in WdgIf.c <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>[?] <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>Move inclusion of Wdg<xxx>.h from wdgIf.h to wdgIf.c. <*>Example of inclusion of Wdg<xxx>.h: <*>#if (WDGIF_NUMBER_OF_DEVICES == 1U) <*>#include "Wdg.h" <*>#endif <*>#if (WDGIF_NUMBER_OF_DEVICES != 1U) <*>#ifdef WDG_INSTANCE0 <*>#if (WDG_INSTANCE0 == STD_ON) <*>#include "Wdg_43_Instance0.h" <*>#endif <*>#endif <*>#ifdef WDG_INSTANCE1 <*>#if (WDG_INSTANCE1 == STD_ON) <*>#include "Wdg_43_Instance1.h" <*>#endif <*>#endif <*>#ifdef WDG_INSTANCE2 <*>#if (WDG_INSTANCE2 == STD_ON) <*>#include "Wdg_43_Instance2.h" <*>#endif <*>#endif <*>#ifdef WDG_INSTANCE3 <*>#if (WDG_INSTANCE3 == STD_ON) <*>#include "Wdg_43_Instance3.h"

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>#endif <*>#endif <*>#ifdef WDG_INSTANCE4 <*>#if (WDG_INSTANCE4 == STD_ON) <*>#include "Wdg_43_Instance4.h" <*>#endif <*>#endif <*>#endif
MCAL-15069	New	New Feature [MCL] Update manuals with information about alignment constraints Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Update manuals with information about alignment constraints: All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively. If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32- byte boundary. Specify for each API which is the alignment constraint for its parameter. Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): Allignment constraints not described in manuals. Expected behavior: Allignment constraints should described in manuals. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update manuals with information about alignment constraints.
MCAL-15080	New	New Feature [RTE] Add support for Crcu driver Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Update RTE plugin to contain the source and header files for Crcu Preconditions: N/A Test Case ID (internal TC that caught the bug) - optional N/A Trigger (not applicable in case of new features): N/A Observed behavior (not applicable in case of new features): N/A Expected behavior: Have the RTE plugin contain the Schm_Crcu .c and .h files

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. Run schm_generator.pl perl script file and push in Git the newly generated .c and .h files for Crcu 2. Update the .mak file of Rte to include the newly generated .c and .h files for Crcu
MCAL-15086	Bug	[PORT] The description of PortPinPcr is not correct.<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>Below is comment from the customer. <*>We were using the MCAL version 'S32K14X_MCAL4_2_BETA_0_9_0' and have configured it using EB tresos tool. There is a configuration field 'PortPinPcr' in tresos. On refering the 'User Manual <*>for S32K14X PORT Driver', we could find that description for both 'PortPinId' and 'PortPinPcr' are the same (sections 4.4.1.2.5 and 4.4.1.2.6). <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the bug) - optional <*>[...] <*>Trigger (not applicable in case of new features): <*>[...] <*>Observed behavior (not applicable in case of new features): <*>The description is not correct. <*>Expected behavior: <*>The description is correct. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>[?] <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>[...]
MCAL-15095	Bug	[WDG] Wrong defines for AR RELEASE version check in Wdg - clone of ENGR00387742<*>Problem detailed description (how to reproduce it): <*>In Wgd.h file, for platforms with only 1 WDG instance the AR Release version macros are defined like this: <*>#define WDG_43_INSTANCE0_AR_RELEASE_MAJOR_VERSION 4 <*>#define WDG_43_INSTANCE0_AR_RELEASE_MINOR_VERSION 0 <*>#define WDG_43_INSTANCE0_AR_RELEASE_REVISION_VERSION 3 <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>#define WDG_AR_RELEASE_MAJOR_VERSION 4 <*>#define WDG_AR_RELEASE_MINOR_VERSION 0 <*>#define WDG_AR_RELEASE_REVISION_VERSION 3
MCAL-15102	Bug	[CAN] Fix Duplicated UUIDs in Can.xdm<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>- There is a CAN's parameter whose UUID is duplicated with the parameter of the another module: <*>UUID : ECUC:fbb4800a-1114-4ef4-9a8b-396befc658cd Can.xdm:4117 Eth.xdm:1005 Fls.xdm:1131 <*>- Change this UUID to make it be unique. <*>Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>NA <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>NA <*>Expected behavior: <*>The UUID is unique. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>NA <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>- Change the UUID: ECUC:fbb4800a-1114-4ef4-9a8b-396befc658cd
MCAL-15105	New	New Feature [FEE] Update default value for FeeEnableUserModeSupport parameter Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Change the default value for FeeEnableUserModeSupport to False as the requirement CPR-MCAL-825.fee : ?The MCAL drivers shall be able to run in a non-privileged processor mode (e.g. User mode). All known related constraints shall be documented. A vendor specific pre-compile boolean configuration parameter <Mdl>EnableUserModeSupport {<MDL>_ENABLE_YSER_MODE_SUPPORT}shall be created for each driver to activate the specific implementation for non-privileged mode. By default, '<Mdl>EnableUserModeSupport' field shall be disabled? Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): NA Expected behavior: The default value of FeeEnableUserModeSupport is False Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): CPR-MCAL-825.fee : (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Change default value of FeeEnableUserModeSupport to False
MCAL-15108	New	New Feature [BASE] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User

Table continues on the next page...

ID	Subtype	Headline and Description
		Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf,

Table continues on the next page...

ID	Subtype	Headline and Description
		Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15111	New	New Feature [BASE] Add errata documents to base module Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): The errata documents must be present in the new environment. Preconditions: [...] Test Case ID (internal TC that caught the bug) - optional [...] Trigger (not applicable in case of new features): [...] Observed behavior (not applicable in case of new features): The errata documents are not present Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add in base/generic/doc a folder named "erratas" in which the errata documents are stored
MCAL-15127	Bug	[WDG] Wrong reference to non-autosar parameter - inactive platforms<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>The definition of parameter WdgClkSrcRef. The destination of the reference is not an AUTOSAR container, therefore the correct definition would be <*><DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/TS_T2D47M10I0R0/Wdg/WdgClockReferencePoint</DESTINATION-REF> <*>the correction shall be done for the parameter in both Fast mode and slow mode. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>The reference is not correct. <*>Expected behavior: <*>The reference is correct. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>NA <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>NA
MCAL-15129	Bug	[CAN] RX and TX MessageBuffers may not be processed when using both the Can FD and interrupt mode <*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>- The current Can driver may have an issue in the following configuration <*> + Can FD is enabled with the separately configuring MB data size. <*>+ and

Table continues on the next page...

ID	Subtype	Headline and Description
		the interrupt processing MBs mode is used <*>- In the interrupt processing MBs mode, the Tx and Rx MBs is divided to process into 2 functions ProcessTx and ProcessRx based on the variable u8FirstTxMBIndex. Where the variable u8FirstTxMBIndex records the index of the first Tx MB in the message RAM area of the hardware. <*>- However, the variable u8FirstTxMBIndex may not be trusted in the above configuration because the Tx and Rx MBs may be mixed in the RAM area. <*>Preconditions: <*>+ Can FD is enabled with the separately configuring MB data size. <*>+ and the interrupt processing MBs mode is used <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>NA <*>Expected behavior: <*>- All MBs are processed correctly <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>NA <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>NA
MCAL-15162	Bug	[ADC] Remove exclusive area in adc_getgroupstatus<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>When calling function Adc_GetGroupStatus continously, the program enters in exclusive 002 and exits immediately. It also clears and removes EEI bit, and get stuck after a lot of loops. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the bug) - optional <*>tc_fnc_adc_00215, test suite Adc_TS_EqAPI_008 <*>Trigger (not applicable in case of new features): <*>[...] <*>Observed behavior (not applicable in case of new features): <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>[?] <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>[...]
MCAL-15167	Bug	[SPI] The driver need to provide a container name CommonPublishedInformation in Spi.xdm file<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>According PR-MCAL-3120.spi: <*>All modules shall provide a container name CommonPublishedInformation holding the common published information of the module. The container CommonPublishedInformation shall be located in pre-configuration section. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>the Spi driver is missing the "CommonPublishedInformation" node in Spi.xdm file <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>Internal requirement PR-MCAL-3120.spi <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>NA

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15171	New	New Feature [ADC] The Adc_<IPVault>_ChannelLimitCheckingType should be put in generic file instead of IPs specific file Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): The Adc_<IPVault>_ChannelLimitCheckingType should be put in generic file instead of IPs specific file due to its elements are generic for all platform Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): NA Expected behavior: The Adc_<IPVault>_ChannelLimitCheckingType will be moved to Adc_Types.h and named as Adc_ChannelLimitCheckingType Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA
MCAL-15311	New	New Feature [FLS] Update the driver to add support for the QUADSPI hardware (S32K14X) ================= Projects ================ Analyzed for all platforms (Calypso,S32K14X, RaceRunner Ultra, Carcassonne, Treerunner): Yes(Yes/No) Did you review and update, if necessary, the Boards Affected: Yes(Yes/No) Defect type (Runtime, Compile time, Link time, Configuration, Documentation): NA(NA for NewFeature) == Impacted Artifacts: ? Safety features impacted: No(Yes/No/NA) ? Requirements Analysis (SWRA) : Yes(Yes/No) ? UML Design: Yes(Yes/No) ? FMEA Report: No (Yes/No) ? N/A for QM platforms ? Source Code: Yes(Yes/No) ? Integration/User Manual: Yes(Yes/No) Is this change (CR) testable? Yes(Yes/No) Root Cause Details: NA Proposed Solution: Create driver for QuadSPI IP, create IPV_QSPI component. Update FLS driver in order to use external QSPI flash sectors. Ticket implemented first on S32V232, based on S32V234RM_Rev1.2.
MCAL-15327	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[PORT] Add control for enabling/disabling User Mode Support in Port plugin, default disabled, readonly=true Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Add control for enabling/disabling User Mode Support in Port plugin, default disabled, readonly=true Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): NA Expected behavior: NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): Quality (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add control for enabling/disabling User Mode Support in Port plugin, default disabled, readonly=true
MCAL-15333	New	New Feature [DIO] Add control for enabling/disabling User Mode Support in Dio plugin, default disabled, readonly=true Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Add control for enabling/disabling User Mode Support in Dio plugin, default disabled, readonly=true Preconditions: NA Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): NA Observed behavior (not applicable in case of new features): NA Expected behavior: NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source (in case of new features): Quality (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add control for enabling/disabling User Mode Support in Dio plugin, default disabled, readonly=true

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15350	New	New Feature [MCU] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15352	New	New Feature [SPI] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15353	New	New Feature [I2C] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15361	New	New Feature [CAN] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S)

Table continues on the next page...

ID	Subtype	Headline and Description
		- have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15366	New	New Feature [ICU] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in thttp://sw-jira.freescale.net/browse/MCAL-15366#he list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15373	New	New Feature [I2C] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15378	New	New Feature [FLS] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for

Table continues on the next page...

ID	Subtype	Headline and Description
		IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15381	New	New Feature [CAN] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15385	New	New Feature [GPT] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units

Table continues on the next page...

ID	Subtype	Headline and Description
		Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15389	New	New Feature [MCU] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15390	New	New Feature [LIN] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the

Table continues on the next page...

ID	Subtype	Headline and Description
		required configuration.
MCAL-15391	New	New Feature [SPI] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15392	New	New Feature [I2C] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15393	New	New Feature [ADC] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for

Table continues on the next page...

ID	Subtype	Headline and Description
		enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15394	New	New Feature [PORT] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15395	New	New Feature [DIO] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from

Table continues on the next page...

ID	Subtype	Headline and Description
		MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15396	New	New Feature [MCL] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15397	New	New Feature [FLS] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the

Table continues on the next page...

ID	Subtype	Headline and Description
		required configuration.
MCAL-15398	New	New Feature [FEE] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15399	New	New Feature [WDG] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15400	New	New Feature [CAN] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for

Table continues on the next page...

ID	Subtype	Headline and Description
		enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-15413	New	New Feature [PORT] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-15414	New	New Feature [DIO] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example:

Table continues on the next page...

ID	Subtype	Headline and Description
		- HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-15415	New	New Feature [MCL] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-15416	New	New Feature [FLS] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature);

Table continues on the next page...

ID	Subtype	Headline and Description
		- DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-15428	New	New Feature [LIN] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15429	New	New Feature [SPI] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from

Table continues on the next page...

ID	Subtype	Headline and Description
		1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15430	New	New Feature [I2C] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15431	New	New Feature [ADC] Errors reported by configuration tolling should reference XDM parameters names

Table continues on the next page...

ID	Subtype	Headline and Description
		Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15432	New	New Feature [PORT] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15433	New	New Feature [DIO] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15435	New	New Feature [FLS] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder.

Table continues on the next page...

ID	Subtype	Headline and Description
		A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15436	New	New Feature [FEE] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15437	New	New Feature [WDG] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on)

Table continues on the next page...

ID	Subtype	Headline and Description
		In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15438	New	New Feature [CAN] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-15447	New	New Feature [LIN] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1

Table continues on the next page...

ID	Subtype	Headline and Description
		(if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \
		Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15453	New	New Feature [MCL] Move FTM common files to GIT Detailed description (how to reproduce it): Add FTM common file to MCL: Ftm_Common_Types.h Reg_eSys_Ftm.h Ftm_Common.c Version to be adapted with G3: BLN_IPV_FTM_SMCAL_4.0_01.12.01 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: NA Expected behavior: NA Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): -Add IPV_FTM folder in \drivers\AutoSAR\mcl\ip -Update make file -Update ifelse(M4_SRC_AR_RELEASE_REVISION...
MCAL-15455	Bug	[SPI] The driver cannot transfer data in SPI_POLLING_MODE if use SpiDataShiftEdge=LEADING and select Continuous CS<*>Detailed description (how to reproduce it): <*>Spi driver has problem in SPI_POLLING_MODE without DMA or interrupt mode without DMA when use SpiDataShiftEdge=LEADING(Data is changed on the leading edge of SCK and captured on the following edge) in continue CS mode. WCF flag cannot set when write first frame into TDR register. <*>This issue occurs because the WCF flag is not working properly in the case of Master mode with SPI_POLLING_MODE without DMA or interrupt request without DMA and use SpiDataShiftEdge is LEADING and CS continuous. <*>So, the SPI driver cannot transfer successfully in that case. <*>Preconditions: <*>- use SPI_POLLING_MODE or interrupt mode (AsyncTransmit in non-DMA) <*>- SpiDataShiftEdge=LEADING <*>- Select Continue CS <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Transfer cannot finish. <*>Expected behavior: <*>Transfer is successful <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Use interrupt of TDF flag instead of WCF flag.
MCAL-15456	New	New Feature [SPI] Some containers node need to update PreCompile for All Variants NewWork Description: Some containers node need to update only support PreCompile: "SpiSequence", "SpiChannel", "SpiChannelList", "SpiJob", "SpiExternalDevice". All of them have no statement about Post-Build Variant Multiplicity being true or False. if not defined otherwise, the Post-Build Variant Multiplicity should be

Table continues on the next page...

ID	Subtype	Headline and Description
		false. Because SpiSequenceId, JobId, ChannelId have requirement requires symbolic name for them and the variant handling means that we have the same application, but with change inside the configuration. The name and IDs of them should the same in multiple post build variants Requirement source: configuration in multiple post build variants. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): the Post-Build Variant Multiplicity should be false. the ImplementationConfigClass shall be PreCompile for All Variants.
MCAL-15457	Bug	[CAN] Wrong reference to non-autosar parameter<*>Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): <*>The definition of parameter CanRAMBlockRef. The destination of the reference is not an AUTOSAR container, therefore the correct definition would be <*><DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/TS_T2D47M10I0R0/Can/CanConfigSet/CanController/CanRAMBlock</DESTINATION-REF> <*>the correction shall be done for the parameter in both Fast mode and slow mode. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the bug) - optional <*>NA <*>Trigger (not applicable in case of new features): <*>NA <*>Observed behavior (not applicable in case of new features): <*>The reference is not correct. <*>Expected behavior: <*>The reference is correct. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Requirement source (in case of new features): <*>NA <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>NA
MCAL-15458	Bug	[WDG] WdgExternalTriggerCounterRef description is confusing<*>Initial Description: <*>Customer questions about the word "either" in "Reference to either - a GptChannelReference sed for the watchdog servicing routine implementation". Is there only GptChannelReference is used for WDG servicing routine or some other source? <*>(Note: After the CR type is decided, will remain only one section from the two listed below, the other one will be deleted) <*>Problem detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Observed behavior: <*>[...] <*>When can it be observed? (at configuration time, at runtime, at compile time?) <*>[...] <*>Expected behavior: <*>[...] <*>Reported release baseline: <*>[...] <*>Proposed solution (Optional): <*>[...] <*>NewWork Classification: (internal task, improvement, feature request) <*>[...] <*>In case for feature request, does a TWG ticket exist (cloned or linked)? TBC (Yes/No) <*>NewWork Description: <*>[?] <*>Expected behavior: <*>[?] <*>Requirement source: <*>[?] <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
MCAL-15497	Bug	[PWM] Add precompile elements to xdm for S32K <*>1. Channels enabling define is precompile data, used to check conflict resourse between ICU, PWM, OCU. Thus, it should be gotten from precompile elements of tresos. <*>2. PwmEnablePhaseShift should be created in PwmGeneral <*>Solution: <*>1. Add a node Channel Enable checking in PwmHwIntrruptConfigList, update

Table continues on the next page...

ID	Subtype	Headline and Description
		Pwm_Cfg.h to get data properly. <*>2. Add PwmGeneral in PwmGeneral. Also update Pwm_Cfg.h <*>
MCAL-15501	New	New Feature [Base] ResumeAllInterrupts() from Mcal_Arm.h is defined twice the ResumeAllInterrupts() macro is defined twice in MCAL_Arm.h
MCAL-15519	Bug	[MCU] Mcu.mak is referring to wrong driver xdm file during compile plugin for ASR 4.2.2<*>Detailed description (how to reproduce it): <*>The Mcu.mak is using ASR_REL_4_2_REV_0001 to separate the implementation for ASR 4.0 and 4.2 as below: <*>ifeq ($ (AR_RELEASE_REVISION),ASR_REL_4_2_REV_0001) <*>else <*>#endif <*>Since we have upgraded to ASR 4.2.2, the above code come to be invalided when compiling plugin for ASR 4.2.2. <*>Preconditions: <*>Compile plugin for ASR 4.2.2 <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>The code should be: <*>ifeq ($ (AR_RELEASE_REVISION),ASR_REL_4_0_REV_0003) <*>else <*>endif
MCAL-15520	Bug	[ADC] Build fail in case of transfer type is DMA and AdcInterruptEnable is not enabled<*>Detailed description (how to reproduce it): <*>In case of AdcInterruptEnable is not enabled and DMA transfer is configured, some functions do not built: <*>Adc_Adc12bsarv2_DmaTransferComplete0 <*>Adc_Adc12bsarv2_DmaTransferComplete1 <*>It will raise error at build when that function configured as mcl notification. <*>The workaround for that is we can enable AdcInterruptEnable in any case (both interrupt and dma transfer) for all of ADC Unit <*>Preconditions: <*>- ADC Unit 0 and 1 used <*>- Transfer type is DMA <*>- AdcInterruptEnable disabled <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>failed at build because missing declare for Adc_Adc12bsarv2_DmaTransferComplete0 and Adc_Adc12bsarv2_DmaTransferComplete1 <*>Expected behavior: <*>TBD <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>TBD
MCAL-15522	New	New Feature [CAN] Restructure UM and IM using general aproach From ticket CR:ENGR00385868 for restructure the UM/IM to compatible with git limitation. This ticket also introduced general structure for all modules. However, due to shortage of time, Can UM/IM were not able to update the structure during that period. This ticket is created to update again the structure of UM/IM follow the general structure.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15523	New	New Feature [ADC] Increase ADC performance for stopping conversions by configuration parameter AdcBypassConsistencyLoop Add the node AdcBypassConsistencyLoop to xdm file to increase performance of Adc. Incase of AdcBypassConsistencyLoop enabled then the stopping conversion will bypass the loop checking timeout. IP layer already support.
MCAL-15524	Bug	[SPI] Missing endif of ifndef SPI_PRECOMPILE_SUPPORT in Spi_Lpspi.h<*>Detailed description (how to reproduce it): <*>Missing "#endif" of "#ifndef SPI_PRECOMPILE_SUPPORT" in Spi_Lpspi.h <*>Preconditions: <*>Build test case use plugin with tag INT_SPI_S32K14X_001 on GIT <*>Test Case ID (internal TC that caught the defect) - optional <*>all test case <*>Observed behavior: <*>Missing "#endif" of "#ifndef SPI_PRECOMPILE_SUPPORT" in Spi_Lpspi.h <*>Expected behavior: <*>no error appear when build test case <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Check if not define SPI_PRECOMPILE_SUPPORT is superfluous in Spi_LPspi.h <*>so, we need remove it.
MCAL-15640	New	New Feature [GPT] Necessary config pointer is not available in <Module>.h NewWork Description: EcuM module executes the Init function of module. Therefore only <module>.h file will be included into generated Source code. If configuration variant PostBuild is used in <module> <module>.h file does not include the <module>_PBcfg.h file. The necessary config pointer is not available. [SWS_Mcu_00215]: The type definitions for Mcu_Lcfg.c and Mcu_PBcfg.c are located in the file Mcu.h. () Workaround: ~~~~~~ Workaround for Tresos 19.x4 Do not use configuration variant PostBuild in <module>. Please see suggestion to put this code to <Module>_Cfg.h as following: [!IF "IMPLEMENTATION_CONFIG_VARIANT = 'VariantPostBuild'"!] #include "Port_PBcfg_[!IF "var:defined('postBuildVariant')"!] [!"$postBuildVariant"!][!ENDIF!].h" [!ENDIF!] For Port_PBcfg.h, #include "Port_Cfg.h" need to be removed. See more attachment files. ~~~~~~ Expected behavior: N/A Requirement source: ASR SWS, Customer request Proposed solution (Optional): <Module>_PBCfg.h shall be removed from the plugin.

Table continues on the next page...

ID	Subtype	Headline and Description
		The configuration export will be done through <Module.h> that includes <Module>_Cfg.h. In <Module>_Cfg.h the following macro will be used to generate the configuration pointer names. [!MACRO "<Module>ExportDriverConfiguration"!] [!NOCODE!][!// [!VAR "configName" = "as:name(<ModuleConfigurationContainerName>)"!][!// /* <comment misra violation 19.4> */ [!CODE!][!// #define <MODULE>_CONF_PB \ [!ENDCODE!][!// [!IF "var:defined('postBuildVariant')"!][!// [!VAR "variantIndex"="0"!][!// [!VAR "variantNumber"="variant:size()"!][!// [!LOOP "variant:all()"!][!VAR "variantIndex"="$variantIndex + 1"!][!// [!// [!CODE!][!WS4!]extern CONST(<ModuleConfigurationType>, <MODULE>_CONST) [!"$configName"!]_[!"."!];[!IF "$variantIndex < $variantNumber"!]\[!ENDIF!][!CR!][!ENDCODE!]
		[!ENDLOOP!][!// [!ELSE!][!// [!CODE!][!WS4!]extern CONST(<ModuleConfigurationType>, <MODULE>_CONST) [!"$configName"!];[!CR!][!ENDCODE!] [!ENDIF!][!// [!ENDNOCODE!][!// [!ENDMACRO!] e.g for calling the macro [!IF "((IMPLEMENTATION_CONFIG_VARIANT != 'VariantPostBuild') and (variant:size()>1)) or (IMPLEMENTATION_CONFIG_VARIANT = 'VariantPostBuild')"!] [!CALL "McuExportDriverConfiguration"!] [!ENDIF!] In <Module>.h the define should be expanded: #if (<MODULE>_PRECOMPILE_SUPPORT == STD_OFF) <MODULE>_CONF_PB #endif /* (<MODULE>_PRECOMPILE_SUPPORT == STD_OFF) */ The expected output is: For IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile and the number of variants <= 1: <MODULE>_CONF_PB should not be generated For IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile and the number of variants > 1: <MODULE>_CONF_PB should be generated and will contain all the names of the configuration pointers For IMPLEMENTATION_CONFIG_VARIANT = VariantPostBuild and the number of variants = 0: <MODULE>_CONF_PB should be generated and will contain the name of the configuration pointer (with no variant in the name) For IMPLEMENTATION_CONFIG_VARIANT = VariantPostBuild and the number of variants > 0: <MODULE>_CONF_PB should be generated and will contain all the names of the configuration pointers

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15644	New	New Feature [GPT] Check parameters passed to Initialization functions NewWork Description: Implement requirement SWS_BSW_00050 [SWS_BSW_00050] Check parameters passed to Initialization functions - If the parameter checking for the Initialization function is enabled (SWS_BSW_00049), the Configuration pointer argument shall be checked with the following conditions: In the supportedConfigVariants VariantPreCompile and VariantLinkTime if only one configuration variant set is used, the initialization function does not need nor evaluate the passed argument. Thus the Configuration pointer shall have a NULL_PTR value. In the supportedConfigVariant VariantPostBuild or if multiple configuration variant sets are used, the initialization function requires the passed argument. Thus the Configuration pointer shall be different from NULL_PTR. If these conditions are not satisfied, a Development error with type "Invalid configuration set selection" shall be reported to Development Error Tracer (Det), see SWS_BSW_00151.] (SRS_BSW_00414, SRS_BSW_00400, SRS_BSW_00438) Requirement source: ASR SWS (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Generate the defined <DRIVER>_PRECOMPILE_SUPPORT in <Driver>_Cfg.h [!IF "IMPLEMENTATION_CONFIG_VARIANT='VariantPreCompile' or IMPLEMENTATION_CONFIG_VARIANT='VariantLinkTime' "!] [!IF "variant:size()<=1"!] <DRIVER>_PRECOMPILE_SUPPORT [!ENDIF!] [!ENDIF!] In <Driver>.c the code that contained parameter checking for init function shall be: #ifdef <DRIVER>_PRECOMPILE_SUPPORT if (NULL_PTR != pConfigPtr) #else if (NULL_PTR == pConfigPtr) #endif /* <DRIVER>_PRECOMPILE_SUPPORT */
MCAL-15663	New	New Feature [PWM] Necessary config pointer is not available in <Module>.h NewWork Description: EcuM module executes the Init function of module. Therefore only <module>.h file will be included into generated Source code. If configuration variant PostBuild is used in <module> <module>.h file does not include the <module>_PBcfg.h file. The necessary config pointer is not available. [SWS_Mcu_00215]: The type definitions for Mcu_Lcfg.c and Mcu_PBcfg.c are located in the file Mcu.h. () Workaround: ~~~~~~ Workaround for Tresos 19.x4 Do not use configuration variant PostBuild in <module>.

Table continues on the next page...

ID	Subtype	Headline and Description
		Please see suggestion to put this code to <Module>_Cfg.h as following: [!IF "IMPLEMENTATION_CONFIG_VARIANT = 'VariantPostBuild'"!] #include "Port_PBcfg_[!IF "var:defined('postBuildVariant')"!] [!"$postBuildVariant"!][!ENDIF!].h" [!ENDIF!] For Port_PBcfg.h, #include "Port_Cfg.h" need to be removed. See more attachment files. ~~~~~~ Expected behavior: N/A Requirement source: ASR SWS, Customer request Proposed solution (Optional): <Module>_PBCfg.h shall be removed from the plugin. The configuration export will be done through <Module.h> that includes <Module>_Cfg.h. In <Module>_Cfg.h the following macro will be used to generate the configuration pointer names. [!MACRO "<Module>ExportDriverConfiguration"!] [!NOCODE!][!// [!VAR "configName" = "as:name(<ModuleConfigurationContainerName>)"!][!// /* <comment misra violation 19.4> */ [!CODE!][!// #define <MODULE>_CONF_PB \\ [!ENDCODE!][!// [!IF "var:defined('postBuildVariant')"!][!// [!VAR "variantIndex"="0"!][!// [!VAR "variantNumber"="variant:size()"!][!// [!LOOP "variant:all()"!][!VAR "variantIndex"="$variantIndex + 1"!][!// [!// [!CODE!][!WS4!]extern CONST(<ModuleConfigurationType>, <MODULE>_CONST) [!"$configName"!]_[!"."!];[!IF "$variantIndex < $variantNumber"!]\\[!ENDIF!][!CR!][!ENDCODE!]
		[!ENDLOOP!][!// [!ELSE!][!// [!CODE!][!WS4!]extern CONST(<ModuleConfigurationType>, <MODULE>_CONST) [!"$configName"!];[!CR!][!ENDCODE!] [!ENDIF!][!// [!ENDNOCODE!][!// [!ENDMACRO!] e.g for calling the macro [!IF "((IMPLEMENTATION_CONFIG_VARIANT != 'VariantPostBuild') and (variant:size()>1)) or (IMPLEMENTATION_CONFIG_VARIANT = 'VariantPostBuild')"!] [!CALL "McuExportDriverConfiguration"!] [!ENDIF!] In <Module>.h the define should be expanded: #if (<MODULE>_PRECOMPILE_SUPPORT == STD_OFF) <MODULE>_CONF_PB #endif /* (<MODULE>_PRECOMPILE_SUPPORT == STD_OFF) */ The expected output is: For IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile and the number of variants <= 1: <MODULE>_CONF_PB should not be generated For IMPLEMENTATION_CONFIG_VARIANT = VariantPreCompile and the

Table continues on the next page...

ID	Subtype	Headline and Description
		number of variants > 1: <MODULE>_CONF_PB should be generated and will contain all the names of the configuration pointers For IMPLEMENTATION_CONFIG_VARIANT = VariantPostBuild and the number of variants = 0: <MODULE>_CONF_PB should be generated and will contain the name of the configuration pointer (with no variant in the name) For IMPLEMENTATION_CONFIG_VARIANT = VariantPostBuild and the number of variants > 0: <MODULE>_CONF_PB should be generated and will contain all the names of the configuration pointers
MCAL-15664	New	New Feature [PWM] Check parameters passed to Initialization functions NewWork Description: Implement requirement SWS_BSW_00050 [SWS_BSW_00050] Check parameters passed to Initialization functions - If the parameter checking for the Initialization function is enabled (SWS_BSW_00049), the Configuration pointer argument shall be checked with the following conditions: In the supportedConfigVariants VariantPreCompile and VariantLinkTime if only one configuration variant set is used, the initialization function does not need nor evaluate the passed argument. Thus the Configuration pointer shall have a NULL_PTR value. In the supportedConfigVariant VariantPostBuild or if multiple configuration variant sets are used, the initialization function requires the passed argument. Thus the Configuration pointer shall be different from NULL_PTR. If these conditions are not satisfied, a Development error with type "Invalid configuration set selection" shall be reported to Development Error Tracer (Det), see SWS_BSW_00151.] (SRS_BSW_00414, SRS_BSW_00400, SRS_BSW_00438) Requirement source: ASR SWS (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Generate the defined <DRIVER>_PRECOMPILE_SUPPORT in <Driver>_Cfg.h [!IF "IMPLEMENTATION_CONFIG_VARIANT='VariantPreCompile' or IMPLEMENTATION_CONFIG_VARIANT='VariantLinkTime' "!] [!IF "variant:size()<=1"!] <DRIVER>_PRECOMPILE_SUPPORT [!ENDIF!] [!ENDIF!] In <Driver>.c the code that contained parameter checking for init function shall be: #ifdef <DRIVER>_PRECOMPILE_SUPPORT if (NULL_PTR != pConfigPtr) #else if (NULL_PTR == pConfigPtr) #endif /* <DRIVER>_PRECOMPILE_SUPPORT */
MCAL-15688	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[GPT] Update xdm to match IRQ names since new silicon version cut 2.0 Update xdm check for channels used.
MCAL-15775	New	New Feature [CRCU] Add support for ASR 4.2.2 standard Detailed description : Add support for ASR 4.2.2 Standard: - update driver configuration according to ASR 4.2.2 standard - update development test configurations for ASR 4.2
MCAL-15799	Bug	[CAN] Can_Flexcan_ChangeBaudrate always clear the memory area of 96 MBs<*>Detailed description (how to reproduce it): <*>+ In the initialization period, Can_Flexcan_ChangeBaudrate always clear the memory area of 96 MBs. Some platforms have controllers which have smaller the number of MBs (for example S32R274 has the CAN1 which has the max MB as 64), this situation may be wrongly in access into the unavailable memory area. <*>+ The driver code is implementing as following: <*>u32TempAddr = 0x0080U; <*>while(u32TempAddr < 0x0680U) <*>{ <*>/* @violates @ref Can_Flexcan_c_REF_2 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ <*>/* @violates @ref Can_Flexcan_c_REF_9 Violates MISRA 2004 Advisory Rule 11.3, A cast should not be performed */ <*>REG_WRITE32(CAN_GET_BASE_ADDRESS(u8HwOffset) + u32TempAddr, (uint32)0U); <*>u32TempAddr = u32TempAddr + 4U; <*>} <*>=> The driver code should clear the memory area based on the maximum MB of the controller configured. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>u32TempAddr = 0x0080U; /* 0x0080U -> FLEXCAN_MB_OFFSET_U32 */ <*>while(u32TempAddr < 0x0680U) /* 0x0680U -> FLEXCAN_MB_OFFSET_U32 + u8MessageBufferControllerSize[CanStatic_pControlerDescriptors[Controller].u8ControllerOffset] * 16 */ <*>{ <*>/* @violates @ref Can_Flexcan_c_REF_2 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ <*>/* @violates @ref Can_Flexcan_c_REF_9 Violates MISRA 2004 Advisory Rule 11.3, A cast should not be performed */ <*>REG_WRITE32(CAN_GET_BASE_ADDRESS(u8HwOffset) + u32TempAddr, (uint32)0U); <*>u32TempAddr = u32TempAddr + 4U; <*>}
MCAL-15800	New	New Feature [MCU] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]"

Table continues on the next page...

ID	Subtype	Headline and Description
		caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15801	New	New Feature [LIN] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15802	New	New Feature [SPI] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/>

Table continues on the next page...

ID	Subtype	Headline and Description
		</a:da>
MCAL-15803	New	New Feature [I2C] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15804	New	New Feature [ADC] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15809	New	New Feature [FLS] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file.

Table continues on the next page...

ID	Subtype	Headline and Description
		For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15811	New	New Feature [WDG] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15817	New	New Feature [ICU] Check validity during configuration time for reference parameters Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference"

Table continues on the next page...

ID	Subtype	Headline and Description
		Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-15825	New	New Feature [GPT] Update exclusive areas for S32K Exclusive areas report needs update for Kinetis
MCAL-15828	Bug	[GPT] Build failed because of template file<*>Detailed description (how to reproduce it): <*>In Gpt_PluginMacros_42.m. there are some lines of code as below: <*>[...] <*>[!CODE!][!// <*>(uint8)(0U), /* LPTMR Clock divider, LPTMR is not USED */ <*>(Gpt_PrescalerType)(0U), /* LPTMR Clock Select, LPTMR is not USED */[!ENDCODE!][!// <*>[!ENDIF!][!// <*>[!CODE!][!// <*>#if (GPT_SET_CLOCK_MODE == STD_ON) <*>[!ENDCODE!] <*>After code is generated, these line will become: <*>(Gpt_PrescalerType)(0U), /* FTM Clock divider, FTM is not used */ (uint8)(0U), /* LPTMR Clock divider, LPTMR is not USED */ <*>(Gpt_PrescalerType)(0U), /* LPTMR Clock Select, LPTMR is not USED */#if (GPT_SET_CLOCK_MODE == STD_ON) <*>This behavior will make errors at build: <*>"#" not expected here <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>Adc_TSe_EqAPI_005 <*>Observed behavior: <*>Failed at build. <*>Expected behavior: <*>Please check aslo the output from template following coding rules. <*>Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>The [!CODE!][!// and [!ENDCODE!][!// at section of code In Gpt_PluginMacros_42.m file: <*>[!CODE!][!// <*>(uint8)(0U), /* LPTMR Clock divider, LPTMR is not USED */ <*>(Gpt_PrescalerType)(0U), /* LPTMR Clock Select, LPTMR is not USED */[!ENDCODE!][!// <*>should be put in the same column. <*>[!CODE!][!// <*>(uint8)(0U), /* LPTMR Clock divider, LPTMR is not USED */ <*>(Gpt_PrescalerType)(0U), /* LPTMR Clock Select, LPTMR is not USED */ <*>[!ENDCODE!][!// <*>
MCAL-15833	New	New Feature [CAN] Improve implementation for the driver code Detailed description (how to reproduce it): Some improvement can be performed for the Can driver code, including: 1. Can_Flexcan_Write: should not access CAN_MCR to check using Pretended Networking, or we must call MCAL_Trusted_function when Can_Write is called. 2. - Remove u8ControllerIdMapping because it is not used driver. - eObjectTypeMapping is used Can.c, but we can get the similar information from Can_MBConfigObjectType instead of eObjectTypeMapping. 3. Should use "if ... else" instead of "<condition> ? value_true : value_false" . 4. In the code generator of ASR 4.2, should use the [!INDENT "..."!] syntax instead of [!WS "..."!] for indentation. 5. Review the CAN_MCR configuration in Can_init and Can_ChangeBaudrate to prevent the duplicate configuration. 6. Should divide the 64 bit variables (CanIcomSignalMask and CanIcomSignalValue) into two 32bits-vairables 7. Don't need to check node:exit for the nodes which is not optional, for

Table continues on the next page...

ID	Subtype	Headline and Description
		example CanIcomSignalMask, ... Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-15862	New	New Feature [CAN] Support OsCounter for timeout to exit blocking point Below is a request from our customer. They expect the CAN driver will use OsCounter for timeout to exit the blocking point as AUTOSAR specification (CAN398, CAN281, CAN113, CAN234, CAN431, CAN397, CAN371,CAN372) . My opinion is to add a new feature to implement this beside current solution (using number of loop), and update the driver user manual deviation requirement accordingly. {quote}I am referring to the MCAL drivers. We observed that referencing an OS counter in the field CanOSCounterReference does not have any effect on the generated code. Looking into the actual driver implementation we noticed that instead of actual time measurements, timeouts are calculated based on loop counters. This can be very inconsistent since the actual timeout can depend on the CPU speed or compiler optimization. that's a trial and error because we can't connect the actual value to something meaningful, we can only base our settings on observations like: this loop should not execute more than 20 times. The problem with that is the uncertainty, what does that mean in terms of seconds/microseconds. Some projects may use faster clock speeds so they would also need to tune their value, again based only on observations. My point is: we are expecting that this will be corrected in the future. {quote}
MCAL-15905	New	New Feature [FLS] Use TresosPluginBuilder M4 tag for platform specific compiler In files Fls_Cfg.c, Fls_PBcfg.c, Fls_Ac.c, Fls_PBcfg_42.c and Fls_LLD_Code.m4, it is used the M4_SRC_COMPILER_6 keyword for differentiating functions declared for Linaro(GCC) compiler, which requires using the "__attribute__" keyword instead of linker section placement. Currently, the M4_SRC_COMPILER_6 keyword is defined in the Fls.mak file. In "AMPT-220 [TPB] Add M4 tag for platform specific compiler define" this define has been added in the Tresos Plugin Builder, visible at FLS driver level, thus being no need to keep these defines in the Fls.mak file. Proposed solution: - remove "COMPILER_x" definitions from Fls.mak - keep the same M4_SRC_COMPILER_6 tag in all source files, no change needed - check that proper define is generated after plugin generation, with the correct

Table continues on the next page...

ID	Subtype	Headline and Description
		Linaro and platform name.
MCAL-15906	New	New Feature [ADC] Add support to use 16 SC registers on S32K144 On Kinetis: - Each ADC supports 16 SC registers: ADC_SC1A -> ADC_SC1P. - Each PDB supports 2 channels: PDB_CH0 and PDB_CH1 Old chip version, we can only use PDB_CH0 to trigger 8 SC registers from ADC_SC1A to ADC_SC1H and PDB_CH1 doesn't work. With the new version chip, PDB_CH1 works and it can trigger 8 SC registers from ADC_SC1I to ADC_SC1P. It's nice when we can use all of 16 SC registers at the same time to improve performance for ADC. One notice that PDB_CH0 and PDB_CH1 cannot trigger ADC at the same time. So the idea for to implement that: - Back to back mode: PDB channel 0: Configured as back to back for all of channels. PDB channel 1: The first channel have to used the delay (I think about the maximum delay to convert 8 channels). The other channels will configured as back to back. - Delay mode will keep the current implementation.
MCAL-15910	New	New Feature [PORT] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15911	New	New Feature [DIO] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers.

Table continues on the next page...

ID	Subtype	Headline and Description
		For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15935	New	New Feature [MCU] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15936	New	New Feature [LIN] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference

Table continues on the next page...

ID	Subtype	Headline and Description
		manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15937	New	New Feature [SPI] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15938	New	New Feature [I2C] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15939	New	New Feature [ADC] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		- S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15940	New	New Feature [DIO] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15941	New	New Feature [PORT] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15942	New	New Feature [CRCU] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15943	New	New Feature [GPT] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual.

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15944	New	New Feature [PWM] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15945	New	New Feature [ICU] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15946	New	New Feature [FLS] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		- S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15947	New	New Feature [FEE] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15948	New	New Feature [MCL] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15949	New	New Feature [WDG] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15950	New	New Feature [CAN] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual.

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15951	New	New Feature [ETH] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15952	New	New Feature [RESOURCE] Update resource according to new Reference manual Rev.3 for S32K14X RTM 1.0.1 ASR 4.0. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15913 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: The S32K14X 4.0 RTM 1.0.1 release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-15959	Bug	[DIO] "<" character must not be contained in the attribute "true" which associated with an element type "null" <*>Detailed description (how to reproduce it): <*>There is an error in Dio.xdm when compile plugin: <*>The value of attribute "true" associated with an element type "null" must not contain the '<' character. <*>This errors refers to DioPortMask parameter. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-15966	New	New Feature [I2C] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration

Table continues on the next page...

ID	Subtype	Headline and Description
		#ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is neuser moeded for running in de the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15967	New	New Feature [ADC] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15968	New	New Feature [PORT] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15969	New	New Feature [DIO] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter

Table continues on the next page...

ID	Subtype	Headline and Description
		<Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15970	New	New Feature [CRCU] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15972	New	New Feature [FLS] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration

Table continues on the next page...

ID	Subtype	Headline and Description
		#ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15973	New	New Feature [FEE] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fee driver already flow those improvements. It is not needed to update driver.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15975	New	New Feature [CAN] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15977	New	New Feature [ETH] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter

Table continues on the next page...

ID	Subtype	Headline and Description
		<Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-15988	New	New Feature [LIN] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-15990	Bug	[CAN] The CAN_FDCTRL[TDCEN] is not configured<*>Detailed description (how to reproduce it): <*>They claim that setting the checkbox for TDCEN (FDCTRL regsiter) has no effect, i.e. register bit is not set in hardware. <*>Preconditions: <*>CanControllerTxBitRateSwitch is checked. <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>CAN_FDCTRL[TDCEN] is not set when the customer enabled CanControllerTrcvDelayCompensationOffset and configured with non-zero value <*>Expected behavior: <*>CAN_FDCTRL[TDCEN] is set when the customer enabled CanControllerTrcvDelayCompensationOffset and configured with non-zero value. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>In Can_FlexCan.c, if the u32CanControllerTrcvDelayCompensation is not 0, set CAN_FDCTRL[TDCEN] before following line. <*>REG_RMW32(FLEXCAN_FDCTRL(u8HwOffset), FLEXCAN_FDCTRL_TDCOFF_U32,(uint32)((uint32)(pCanControlerDescriptor->pControllerBaudrateConfigsPtr[u8BaudrateIndex].ControllerFD.u32CanControllerTrcvDelayCompensation)<<FLEXCAN_FDCTRL_TDCOFF_OFFSET_U8));

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-15993	Bug	[ADC] PDB could trigger a second (spurious) ADC conversion for a single HW trigger<*>Detailed description (how to reproduce it): <*>In the following scenario: <*>An ADC Hw Triggered group is used together with Adc_SetChannel API, called from group conversion complete notification. <*>After each conversion, the list of channels and delays is changed. <*>The source of triggers is a PWM with low frequency (long period). <*>For the first conversion, the configured delays are small - so the conversion occurs in the beginning of PWM period, notification and Adc_SetChannel are called. <*>The delays are then changed to much larger values; the PDB counter continues to count; <*>In this case, it can happen that the PDB counter has another match to the delay values, thus triggering another Adc conversion for the same HW trigger. <*>Preconditions: <*>see above <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>2 ADC conversions are triggered for 1 HW trigger <*>Expected behavior: <*>only 1 ADC conversion is triggered for 1 HW trigger. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Use PDB modulo register to stop the PDB counter after the group conversion is triggered, preventing a second match even for the same period, in case PDB delays are changed via Adc_SetChannel.
MCAL-15994	Bug	[GPT] Set to normal mode should not reset predef timer<*>Detailed description (how to reproduce it): <*>When GPT driver is being in NORMAL state, call Gpt_SetMode(GPT_NORMAL_MODE), counter value of all predef timer reset to 0 <*>Requirement SWS_Gpt_00392 <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>counter is reset <*>Expected behavior: <*>counter should not be reset <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-16025	Bug	[PWM] Fix violations of Misra rule 10.5 not caught by PC Lint tool<*>Problem detailed description (how to reproduce it):<*>Using GHS Misra checks, customer has found that one line of code in file Dio_Siul2.c is violating rule 10.5.<*>This violation is not reported by PC Lint tool.<*>The violation can be checked by compiling the file individually with GHS and adding the misra check option: "--misra_2004=8.5,8.10,10.5,17.5,20.4,20.5"<*>The line of code breaking the rule 10.5 is:<*>Pwm_Ftm_aNotifToChannelMap[nLogicalChannelIndex] &= ~(PWM_TOF_IRQ_NO_EDGE_NOTIF);<*>Preconditions:<*>Running Misra checks with GHS compiler<*>Test Case ID (internal TC that caught the defect) - optional<*>N/A<*>Trigger:<*>Running Misra checks with GHS compiler<*>Observed behavior:<*>Compilation with GHS cannot be completed.<*>Expected behavior:<*>GHS compilation with option --misra_2004=8.5,8.10,10.5,17.5,20.4,20.5 should complete without errors.<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):<*>Update the line in driver code<*>From<*>Pwm_Ftm_aNotifToChannelMap[nLogicalChannelIndex] &= ~(PWM_TOF_IRQ_NO_EDGE_NOTIF);<*>To<*>Pwm_Ftm_aNotifToChannelMap[nLogicalChannelIndex] &= (uint8)~(PWM_TOF_IRQ_NO_EDGE_NOTIF);

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16071	New	New Feature [SPI] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \ Can_FlexCan_ResetController(Controller); #endif

Table continues on the next page...

ID	Subtype	Headline and Description
		This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16076	New	New Feature [CAN] Buffer size is hard coded instead of a symbolic constant The customer ran the static code analysis and found that the buffer size is hard-coded. 59: Hard-coded buffer size File Scope: Buffer size is hard coded instead of a symbolic constant. Hard-coded buffer size increases maintenance costs and security risks. e.g: inside function Can_FlexCan_ProcessRx, Can_FlexCan_Write the buffer u8DataLengthMax[8], u8DataLengthMin[8] are declared
MCAL-16093	Bug	[FLS] Replace "__GNUC__" define check with platform specific Linaro compiler define<*>Original CQ Reported Baseline: BLN_FLS_SMCAL_4.0_S32K14X_01.00.00<*>Problem detailed description (how to reproduce it):<*>If "?gcc" compiler option is used with a different compiler, than the "__GNUC__" define is generated alongside with the current compiler used.<*>The FLS driver uses the "__GNUC__" define check in order to set LINARO compiler specific parameters. <*>If the above compiler option is used, this newly generated define leads to the scenario where the LINARO specific processor branches are taken instead of the ones intended for the currently used compiler.<*>Depending on setup options,compiler used, etc., this might lead to a compile time error if "__attribute__" keywords are not supported by the compiler defining "-gcc".<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Trigger:<*>Use "?gcc" compiler option with a compiler different than LINARO.<*>Observed behavior:<*>Possible compile time error, if "__attribute__" keywords are not supported by the compiler defining "-gcc"<*>Expected behavior:<*>NA<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>Replace "#ifdef __GNUC__" check with a M4 compiler platform defined in build_env.
MCAL-16144	Bug	[WDG] Can not generate Wdg_Wdog_OffModeSettings_Instance<*>Original CQ Reported Baseline: BLN_WDG_SMCAL_4.2_S32K14X_01.00.01<*>Problem detailed description

Table continues on the next page...

ID	Subtype	Headline and Description
		(how to reproduce it):<*>This is an internal defect, affecting only internal baseline BLN_WDG_SMCAL_4.2_S32K14X_01.00.01.<*>It does not affect any customer release.<*>Because generation code for Wdg_Wdog_OffModeSettings_Instance was moved outside of loop [!LOOP "WdgSettingsConfig/*"!][!//<*>Therefore it issues error when generating as below:<*>Parsing file "C:\tools\tresos_21_0_0\plugins \Wdg_TS_T40D2M10I0R0/generate_PB/src/Wdg_PluginMacros.m", line "105" The XPath-expression "node:value(concat(concat('./',"WdgSettingsOff"),'/ WdgAllowUpdates'))" caused an error: (1818) No value found for object "./ WdgSettingsOff/WdgAllowUpdates"<*>Preconditions:<*>NA<*>Test Case ID (internal TC that caught the defect) - optional<*>NA<*>Trigger:<*>NA<*>Observed behavior:<*>Appear error when when generating<*>Expected behavior:<*>No error when when generating<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>Move <*>[!LOOP "WdgSettingsConfig/ *"!][!//<*>[!IF "node:value(./WdgInstance) = $WdgWdogInstance"!][!//<*>to above Wdg_Wdog_OffModeSettings_Instance.
MCAL-16155	New	New Feature [DIO] Add support to generate UM/IM for distinct versions of Autosar on the same platform For platforms having releases with more than one Autosar version it should be possible to generate UM/IM for each of the ASR version. This means that the following files in the documentation folder should have distinct instances for each of the supported Autosar versions: Files in Common folder: revision_history.xml Files in IM main xml file files_required.xml setting_up_plugins.xml parameters.xml sections_in_memmap_h.xml other_modules_dependencies.xml Files in UM main xml file deviation_from_requirements.xml driver_limitations.xml If for example ASR 4.0 and ASR 4.2 are supported in the same time on a platform, then the file files_required.xml in IM doc folder should appear twice: - files_required.xml - files_required_42.xml Each instance of the file should contain information in sync with the version of Autosar it refers to. Please take a look at the ADC driver for an implementation example.
MCAL-16156	New	New Feature [PORT] Add support to generate UM/IM for distinct versions of Autosar on the same platform For platforms having releases with more than one Autosar version it should be

Table continues on the next page...

ID	Subtype	Headline and Description
		possible to generate UM/IM for each of the ASR version. This means that the following files in the documentation folder should have distinct instances for each of the supported Autosar versions: Files in Common folder: revision_history.xml Files in IM main xml file files_required.xml setting_up_plugins.xml parameters.xml sections_in_memmap_h.xml other_modules_dependencies.xml Files in UM main xml file deviation_from_requirements.xml driver_limitations.xml If for example ASR 4.0 and ASR 4.2 are supported in the same time on a platform, then the file files_required.xml in IM doc folder should appear twice: - files_required.xml - files_required_42.xml Each instance of the file should contain information in sync with the version of Autosar it refers to. Please take a look at the ADC driver for an implementation example.
MCAL-16161	New	New Feature [MCU] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16162	New	New Feature [LIN] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16163	New	New Feature [SPI] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173

Table continues on the next page...

ID	Subtype	Headline and Description
		(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16164	New	New Feature [I2C] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16165	New	New Feature [ADC] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16168	New	New Feature [CRCU] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16169	New	New Feature [GPT] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100

Table continues on the next page...

ID	Subtype	Headline and Description
		s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16170	New	New Feature [PWM] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16171	New	New Feature [ICU] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference

Table continues on the next page...

ID	Subtype	Headline and Description
		manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16172	New	New Feature [FLS] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16175	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[WDG] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16176	New	New Feature [CAN] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): N/A
MCAL-16178	New	New Feature [RESOURCE] Update resource according to new Reference manual Rev.3 for S32K14X ASR 4.2. NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-15836 for more detail about Reference manual. s32k148_lqfp144 s32k148_lqfp176 s32k148_mapbga100 s32k146_lqfp144 s32k146_lqfp100 s32k146_mapbga100 s32k144_lqfp100 s32k144_lqfp64 s32k144_mapbga100 s32k142_lqfp100 s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivatives. For S32K146 derivatives, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-16187	New	New Feature [SPI] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated.

Table continues on the next page...

ID	Subtype	Headline and Description
		3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16192	Bug	[LIN] Correct name for module DET (Default Error Tracer) for ASR 4.2.2<*>Detailed description (how to reproduce it): <*>From ASR 4.2.2, DET has been renamed from "Development Error Tracer" to "Default Error Tracer? <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Correct <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Please see the analysis tab
MCAL-16198	Bug	[CAN] Can controller is sometime in Supervisor Mode after put controller in STOP state<*>Detailed description (how to reproduce it): <*>In progress to change the controller to STOP mode, the soft reset event may be occurred and lead to MCR_SUPV bit is set as 1 (it means the controller is in Supervisor Mode). Thus, the driver will jump to IOVR1 when the User Mode is used and the Can registers are accessed. <*>Preconditions: <*>+ User uses a small value of the timeout duration <*>+ The user mode is enabled <*>+ The Can registers is accessed after change the controller to STOP mode <*>Test Case ID (internal TC that caught the defect) - optional <*>CAN_TC_0020.c <*>Observed behavior: <*>The driver will jump to IOVR1 after accessing the Can registers after changing the controller to STOP mode <*>Expected behavior: <*>The Can registers are able to be accessed successfully in the user mode after the controller is changed to STOP mode. <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16199	New	New Feature [SPI] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers. For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this

Table continues on the next page...

ID	Subtype	Headline and Description
		mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-16200	New	New Feature [MCU] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \

Table continues on the next page...

ID	Subtype	Headline and Description
		Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16203	Bug	[CAN] RX FIFO issue in CAN driver<*>Problem detailed description (how to reproduce it): <*>Target Baseline: BLN_SMCAL_4.0_CALYPSO_RTM_1.0.2. <*>Configs Affected: MPC574XG <*>CanRxFifoTable in RxFifoTableID_PCConfig[CAN_MAXTABLEID_0] generated in Can_Cfg.c (Can_PBcfg.c) is corresponding to CanController Index number but in Can_FlexCan_InitRxFiFo, u8RxFifoTableIdIndex is calculated based on CanControllerId. This problem will, afterwards, select incorrect CanRxFifoTable from RxFifoTableID_PCConfig to write the Table ID and FilterMask if the CanController Index number is not matched to CanControllerId in Can configuration (only happens in the case that more than 1 CanController is configured). <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16214	Bug	[ICU] Validity of ICU...ChannelRef node is over-checked <*>Detailed description (how to reproduce it): <*>After implementation of "feature/MCAL-15817-icu-check-validity-during-configuration" the validity check is done also for reference that is not used forcing the use of some value - but the value used should be not duplicated from other node - this leads to not able to complete some configurations with limited number of references defined. <*>Preconditions: <*>limited number of reference channels defined. use TS_002 to generate the code. <*>Test Case ID (internal TC that caught the defect) - optional <*>TS_002 <*>Observed behavior: <*>error on generation because duplicate reference name used in non-used reference fields. <*>Expected behavior: <*>ignore the non-used reference fields. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>correct the validity checks

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16225	Bug	[CAN] RX FIFO issue in CAN driver<*>Problem detailed description (how to reproduce it):<*>Target Baseline: BLN_SMCAL_4.0_CALYPSO_RTM_1.0.2.<*>Configs Affected: MPC574XG<*>CanRxFifoTable in RxFifoTableID_PCConfig[CAN_MAXTABLEID_0] generated in Can_Cfg.c (Can_PBcfg.c) is corresponding to CanController Index number but in Can_FlexCan_InitRxFiFo, u8RxFifoTableIdIndex is calculated based on CanControllerId. This problem will, afterwards, select incorrect CanRxFifoTable from RxFifoTableID_PCConfig to write the Table ID and FilterMask if the CanController Index number is not matched to CanControllerId in Can configuration (only happens in the case that more than 1 CanController is configured).<*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger:<*>[...] <*>Observed behavior:<*>[...] <*>Expected behavior:<*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>[...]
MCAL-16226	New	New Feature [WDG] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16236	Bug	[FLS] Follow up fix code review against checklist for S32K14x RTM 1.0.1 ASR 4.0<*>NewWork Description: <*>*)Pairs of matching parentheses are either located in the not same line or in the not same column (see in attachment tab) <*>*)An empty line must be placed after the variable declaration and before the code (see in attachment tab) <*>*)All local non-static variables of a function must have the memclass AUTOMATIC (see in attachment tab) <*>uint32 u32Counter; -> VAR(uint32, AUTOMATIC) u32Counter; <*>uint32 u32TimerCounterAbort = FLS_ABT_TIMEOUT_VALUE; -> VAR(uint32, AUTOMATIC) u32TimerCounterAbort = FLS_ABT_TIMEOUT_VALUE; <*>... <*>Please see in attachment <*>Requirement source: <*>N/A <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>Please see in attachment tab
MCAL-16243	New	New Feature [MCU] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16250	New	New Feature [ADC] Improve performance when use the SC1n registers There are many SC1n registers in Kinetis, but not all of them used. Example, - in S32K144 we have 16 SC1 registers from SC1A -> SC1P,

Table continues on the next page...

ID	Subtype	Headline and Description
		- the maximum channels is assigned for group is 8. So we only use from SC1A to SC1H, the others will not be used. we lost many effort to write in to SC1I -> SC1P in Adc_Adc12bsarv2_EndGroupConversion, Adc_Adc12bsarv2_StopConversion, Adc_Adc12bsarv2_StopDmaAndConversion. We can improve performance by the way only write or read the used SC1 registers
MCAL-16254	New	New Feature [MCL] Add support for running from User Mode - inactive platforms NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else

Table continues on the next page...

ID	Subtype	Headline and Description
		#define Call_Can_FlexCan_ResetController(Controller) \ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16256	New	New Feature [MCL] Update Input and register for S32K according to new RM According to new RM S32K14XRM Rev. 3, 03/2017: TRGMUX contains: - New peripheral trigger input - New registers and some registers are not available for all derivative
MCAL-16269	New	New Feature [SPI] Update some defines in Reg_eSys_LPspi.h NewWork Description: Some changes from new Reference Manual(Rev3) for SPI module: - Added the "Table 45-1" in the subchapter "45.1.1 Instantiation Information". - Change size of some fields: + FCR[RXWATER]: from [23-16] to [17-16]. + FCR[TXWATER]: from [7-0] to [1-0]. + FSR[RXCOUNT]: from [23-16] to [18-16]. + FSR[TXCOUNT]: from [7-0] to [2-0]. Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): need to update some defines in Reg_eSys_LPspi.h: - LPSPI_FCR_RESERVED_MASK_U32 - LPSPI_FCR_RXWATER_MASK_U32 - LPSPI_FCR_TXWATER_MASK_U32 - LPSPI_FSR_RESERVED_MASK_U32 - LPSPI_FSR_RXCOUNT_MASK_U32

Table continues on the next page...

ID	Subtype	Headline and Description
		- LPSPI_FSR_TXCOUNT_MASK_U32
MCAL-16271	New	New Feature [I2C] Update condition for M4 tags NewWork Description: Update the way the driver code differences between the ASR4.0.3 and ASR. 4.2.x are handled with M4 tags. e.g. ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_0_REV_0003',`dnl #include "MemMap.h" ',`dnl #include "<Module>_MemMap.h" ')dnl Requirement source: gMRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA
MCAL-16272	New	New Feature [I2C] Add pre-compiler support into the function I2C_Init for ASR 4.2 NewWork Description: Currently, the function I2C_Init() is only support pre-compiler for ASR 4.0.3. => Need to add pre-compiler support into the function I2C_Init for ASR 4.2. The driver need to uses the configuration from the variable I2C_PBCfgVariantPredefined. Requirement source: Developer (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA
MCAL-16273	New	New Feature [BASE] Include CSEC_PRAM_BASE definition Please include CSEC_PRAM_BASE inside BASE module. Right now, the definition can be found inside Reg_eSys_Csec.h. #define CSEC_PRAM_BASE ((uint32)0x14001000)
MCAL-16283	New	New Feature [PORT] Add default configuration xdm file in plugin Investigate each platform in Boards Affected of the ticket to check if JTAG pins shall be explicitely configured by the user. If yes, add a default configuration.xdm file in the Port driver plugin which shall contain the 4 JTAG pins, already configured with the JTAG functionality. For S32K14X platform, add in the default configuration the PTA5 pad configured with Reset_b functionality.

Table continues on the next page...

ID	Subtype	Headline and Description
		Describe in the User Manual the presence of the default configuration in the plugin.
MCAL-16287	New	New Feature [BASE] Add support for the EEP driver Add support for the EEP driver.
MCAL-16291	New	New Feature [LIN] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Check the requirements on Doors and update chapter Deviation from requirements if needed.
MCAL-16294	New	New Feature [RTE] Add support for the EEP driver Add support for the EEP driver.
MCAL-16299	Bug	[DIO] Correct the description of Dio Reverse Port Bits<*>Detailed description (how to reproduce it): <*>There is one attribute in Dio plugin called `Dio Reverse Port Bits?. Looking at its description in Tresos <*>`If this box is checked the bits written to defined ports will be reversed, writing 3 to PORTA with checkbox disabled will set pins 14 and 15, writing 3 to PORTA with checkbox enabled will set pins 0 and 1.? <*>The description above is not correct because the feature behaves exactly the other way arround <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Correct the description <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Change the description to: <*>If this box is checked, the bits written to defined ports will be reversed, meaning that writing 3 to a port with checkbox disabled will set pins 0 and 1 of the port while writing 3 to a port with checkbox enabled will set pins 14 and 15 of the port.
MCAL-16300	New	New Feature [ADC] User Manual for each driver should contain a "How To" configure

Table continues on the next page...

ID	Subtype	Headline and Description
		chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-16304	New	New Feature [PORT] Follow-up ticket to update the information in the resources about pins supporting DSE capability - S32K14X RTM 1.0.1 NewWork Description: The DSE control is available on the high-drive pins (marked as GPIO-HD in the IO Signal Multiplexing Table) In 0N77P (MA512), following are the high-drive pins (These I am seeing from design, the document might have some error): PTB4, PTB5, PTD0, PTD1, PTD15, PTD16, PTE0 and PTE1 In S32K144, following are the high-drive pins: PTA10, PTB4, PTB5, PTB6, PTD0, PTD1, PTD15, PTD16, PTE0, PTE1 and PTE4 Following are the additional high-drive pins. These were added on the recommendation of DFT team. The reasons are mentioned below : PTA10: used as scan-out in test pin muxing PTB6: used as scan-out in test pin muxing PTE4: used as PLL observe in test pin muxing Requirement source: NA Proposed solution (Optional): update the information in the resources about pins supporting DSE capability for 4 derivatives (S32K142, S32K144, S32K146 and S32K148)
MCAL-16310	New	New Feature [ADC] Description of the HW Channel Id should reflect the name given in Reference Manual The issue here reflects the way HW channel IDs are described in both XDM fields (which are visible in the configuration tooling) and the User Manuals for IO capable drivers.

Table continues on the next page...

ID	Subtype	Headline and Description
		For example for Calypso, in case of ADC driver the user can select to configure a HW channel using parameter "AdcChannelId". This parameter can take values from AN0 ... AN15. The pinout described in the Reference Manual uses a different format for HW channel names (depending on the type of the channel: [S]tandard/[P]recision) with different values. In the reference manual, there is a table in chapter *32.1.5 ADC channel mapping* which describes this mapping. The ADC UM references this table in chapter *3.2.1 ADC Units Hardware Channels* however a better solution would be for the UM to have a copy of the table described in the RM since it would make easier for user to access information without using different sources. Same applies for other drivers like PWM, ICU, CAN, LIN, SPI... For PORT and DIO, there is a special case where the mapping between HW pin and the PCR value or the logical channel id value is not properly explained. (i.e. Dio channel value is given by its underlying port-pin value --- by mixing together the Port number and number of the pin in that port)
MCAL-16319	New	New Feature [ADC] Update for Adc_Pdb_ReConfigureDma function in Adc_Pdb_Irq.c the Adc_Pdb_ReConfigureDma can be compiled out with ADC_OPTIMIZE_ONESHOT_HW_TRIGGER this line should be update when 16SC register supported: CurrentChannel = (((CurrentChannel%8U) == 0U) ? (CurrentChannel - 8U) : (CurrentChannel - (CurrentChannel%8U))); to: CurrentChannel = (Adc_ChannelIndexType)ADC_PRE_CHANNEL_INDEX(Adc_aGroupStatus[Group].CurrentChannel);
MCAL-16324	Bug	[CAN] Incorrect warning about bit timing configuration<*>Problem detailed description (how to reproduce it): <*>When the CAN bit time in control field and data field are configured with different number of time quantas. The following warning message appear. <*>"The No of Time Quantas for CAN_CBT (80) is different by No of Time Quantas configurated for CAN_FD (8). <*>This situation can occur a quantization error of up to one time quantum of the arbitration phase, which may be present as a phase error. <*>The No of Time Quantas for CAN_CBT (80) is different by No of Time Quantas configurated for CAN_FD (8). <*>This situation can occur a quantization error of up to one time quantum of the arbitration phase, which may be present as a phase error. <*>The No of Time Quantas for CAN_CBT (80) is different by No of Time Quantas configurated for CAN_FD (8). <*>This situation can occur a quantization error of up to one time quantum of the arbitration phase, which may be present as a phase error." <*>However these messages are not true. <*>According to the reference manual, it was recommended to configure the same value for FPRESDIV (in CAN_FDCBT) and PRESDIV (in CAN_CBT or CAN_CTRL1), so that the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are identical to minimize the chance of error frames on CAN bus. <*>Therefore, the prescaler values should be checked instead of the number o Time Quantas. <*>Below is copy of the NOTE from reference manual. <*>NOTE: To minimize errors when processing FD frames, use the same value for FPRESDIV and <*>PRESDIV (in CAN_CBT or CAN_CTRL1). For more details refer to the first NOTE in section CAN <*>FD frames. <*>NOTE <*>If the length of the time quantum in the nominal bit timing and <*>the length of the time quantum in the data bit timing are not

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>identical, a quantization error of up to one time quantum of the <*>arbitration phase may be present as a phase error. This situation <*>can occur after the switch from arbitration to data phase and <*>will last until the next synchronization event. Thus, the length <*>of the time quantum should be the same in nominal and data bit <*>timing in order to minimize the chance of error frames on the <*>CAN bus, and to optimize the clock tolerance in networks that <*>use FD frames. <*>Preconditions: <*>The bit timming is configured with different number of Time Quantas for nominal bit timing and FD bit timing. <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>The incorrect warning during configuration generation <*>Expected behavior: <*>The warning only appears when the prescalers in (CAN_CBT and CAN_FDCBT) are different. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16327	Bug	[FLS] Build error when using more than one FlsConfigSet in the configuration<*>Detailed description (how to reproduce it): <*>In Fls configuration, we can have more than one instance of FlsConfigSet (FlsConfigSet_0, FlsConfigSet_1?..) <*>And the number of sectors in the each configuration set can be different (FlsConfigSet_0 : 4 sectors, FlsConfigSet_1 : 5 sectors) and the used sectors in each configuration set can be different also. <*>Before we used 1 Fls_Flash_aSectorList0[] definition to physical sector description but this way only good if number of sector are small. <*>New way we will generate 1 array which we config only in EB to Instead using define all the sectors as old way. <*>new way is only true If configuration with 1 FlsConfigSet (FlsConfigSet_0) it is pass but if configuration more than 1 FlsConfigSet it will occurs errors when building (Fls_Flash_aSectorList0[ ] will be redefined and number of element in array are different in FlsConfigSet_0 and (FlsConfigSet_1 ...). <*>Preconditions: <*>One error occurs when building with more than 1 FlsConfigSet. <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>When config with more than 1 configset which test build will fail which Fls_Flash_aSectorList0[ ] is not declared ... <*>Expected behavior: <*>build and run pass with more than 1 configset. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...] -- NewWork Description: <*>N/A <*>Requirement source: N/A <*>Proposed solution (Optional): <*>- Adding a new Fls_Flash_SectorInfoType pointer to Fls_ConfigType. This pointer will be used access Fls_Flash_aSectorList0 of each FlsConfigSet <*>Access Fls_Flash_aSectorList0 using Fls_u32JobSectorIt instead of get index from ConfigSet_X_aFlsPhysicalSector array.
MCAL-16331	New	New Feature [CRCU] Extern declaration of the Crcu configuration structure for pre-compiled mode Declaration of the CRCU configuration structure for pre-compiled mode: Crcu_ConfigPC need to be externed in CDD_Crcu.c.
MCAL-16334	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[FLS] Update the integration manual to better describe the usage of Fls_DsiHandler and the exception handling responsability Detailed description (how to reproduce it): Update the integration manual to better describe the usage of Fls_DsiHandler and the exception handling responsibility: -the FLS driver is responsible for the Fls_DsiHandler implementation -the customer application should use Fls_DsiHandler if the customer exception handling strategy is to allow the memory stack to handle data flash ECC errors -the customer is responsible for the exception handler implementation based on its project specific strategy -the syndrome values should be exported by the FLS driver and visible at the application level Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-16335	Bug	[CAN] When FD is activated without BRS in configuration we still need to configure the second baudrate fields even not needed<*>Detailed description (how to reproduce it): <*>In configuration if FD is enabled, but without BRS field checked, we still need to configure the second baudrate even if this is not needed, with correct values, otherwise we will get errors at generate. <*>This can be avoided by making this fields grayed out. Non editable when BRS not activated. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16337	Bug	[CRCU] Update mak. file in order to be able to generate the signed plugin<*>h1. Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16339	New	New Feature [ADC] Improvement the behavior for group triggered by hardware Currently driver always re-configure the configuration in any case after group complete converse.

Table continues on the next page...

ID	Subtype	Headline and Description

		This is unnecessary when the number channels of group less than ChannelCount > ADC_NUM_SC1n_REGISTER_USED and have no updating for new channels with setchannel
		So I think we can improve performance for hardware triggered group by this way:
		Case 1: Interrupt transfer used:
		in Adc_Adc12bsarv2_UpdateHardwareGroupState function, add this checking before call Adc_Adc12bsarv2_StartHwTrigConversion function like this:
		VAR(Adc_ChannelIndexType, AUTOMATIC) ChannelCount;
		#if (ADC_SETCHANNEL_API == STD_ON)
		ChannelCount = Adc_aRuntimeGroupChannel[Group].ChannelCount;
		#else
		/** @violates @ref Adc_Adc12bsarv2_Irq_c_REF_8 Array indexing shall be the only allowed form of pointer arithmetic */
		ChannelCount = Adc_pCfgPtr->pGroups[Group].AssignedChannelCount;
		#endif
		if((ChannelCount > ADC_NUM_SC1n_REGISTER_USED)
		#if (ADC_SETCHANNEL_API == STD_ON)
		|| ((boolean)TRUE == Adc_aRuntimeGroupChannel[Group].bRuntimeUpdated)
		#endif /* (ADC_SETCHANNEL_API == STD_ON) */
		)
		{
		Adc_Adc12bsarv2_StartHwTrigConversion(Unit, Group);
		}
		Case 2: DMA transfer used:
		Update the Adc_Adc12bsarv2_DmaEndHardwareConv function: incase of the number of channels greater than ADC_NUM_SC1n_REGISTER_USED or channels changed by setchannel api then reconfigure again by Adc_Adc12bsarv2_ConfigureDmaPartialConversion and Adc_Adc12bsarv2_DmaUpdateDaddr. else we only reconfigure DMA setting (source address, destination address, iter etc...) like this:
		if((ChannelCount > ADC_NUM_SC1n_REGISTER_USED)
		#if (ADC_SETCHANNEL_API == STD_ON)
		|| ((boolean)TRUE == Adc_aRuntimeGroupChannel[Group].bRuntimeUpdated)
		#endif /* (ADC_SETCHANNEL_API == STD_ON) */
		)
		{
		u8NumChannel = (uint8)ADC_ADC12BSARV2_GET_LEFT_CHANNEL(ChannelCount);
		/** @violates @ref Adc_Adc12bsarv2_Irq_c_REF_8 Array indexing shall be the only allowed form of pointer arithmetic */
		Adc_Adc12bsarv2_ConfigureDmaPartialConversion(Unit, Group, pGroupPtr, (Adc_HwTriggerTimerType)pGroupPtr->pHwResource[0], u8NumChannel, (boolean)TRUE);
		
		Adc_Adc12bsarv2_DmaUpdateDaddr(Group);
		}
		else
		{
		Adc_Adc12bsarv2_aDmaTcdConfig[Unit].u32saddr = (uint32) ADC12BSARV2_Rn_REG_ADDR32(Unit, (uint32)0);
		/** @violates @ref Adc_Adc12bsarv2_c_REF_9 cast from unsigned long to pointer */
		/** @violates @ref Adc_Adc12bsarv2_c_REF_3 A cast should not be performed between a pointer type and an integral type. */
		Adc_Adc12bsarv2_aDmaTcdConfig[Unit].u32daddr = (uint32) (Adc_pCfgPtr->pGroups[Group].pResultsBufferPtr[Group] +

Table continues on the next page...

ID	Subtype	Headline and Description

		Adc_aGroupStatus[Group].ResultIndex);
		Mcl_DmaConfigTcd(Adc_Adc12bsarv2_aTcdAddress[Unit], &(Adc_Adc12bsarv2_aDmaTcdConfig[Unit]));
		Adc_aGroupStatus[Group].CurrentChannel += ChannelCount;
		}

MCAL-16341	New	New Feature [ICU] Improvements of running from User Mode NewWork Description: Improvements for supporting running from USER MODE 1. A check shall be performed that the driver user mode is enabled only when the MCAL_USER_MODE_SUPPORT is enabled. The check below shall be done in generated configuration file (<Mdl>_Cfg,h) after the <driver>_ENABLE_USER_MODE_SUPPORT is generated from configuration #ifndef MCAL_ENABLE_USER_MODE_SUPPORT #if (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) #error MCAL_ENABLE_USER_MODE_SUPPORT is not enabled. For running <driver> in user mode the MCAL_ENABLE_USER_MODE_SUPPORT needs to be defined #endif /* (STD_ON == <Mdl>_ENABLE_USER_MODE_SUPPORT) */ #endif /* ifndef MCAL_ENABLE_USER_MODE_SUPPORT */ 2. All function calls for configuring register protection shall be guarded using a precompile switch: <DRIVER>_<IP>_REG_PROT_AVAILABLE. The availability of the register protection shall be generated. 3. If no support is needed for running in user mode the paramter <Mdl>EnableUserModeSupport from xdm shall be set on false and made read-only Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16346	New	New Feature [ICU] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM.

Table continues on the next page...

ID	Subtype	Headline and Description
		In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-16347	New	New Feature [PWM] Dependecy to other drivers needs to be documented Please check and update the dependency to other drivers. There are drivers like MCU and MCL whose proper configuration is critical for enablement of other drivers and is not exemplified (or even described) in the overlaying drivers. Examples: - all timer drivers (PWM, ICU, OCU, GPT) may use different MCU clock setups which are critical for obtaining a certain PWM frequency or roll-over period. The dependency to MCU configuration is nowhere described in the PWM/GPT/ICU/OCU drivers - drivers like SPI, CAN, ADC who usually use only a single clock source from MCU should also mention how to configure this clock since it may be possible that on different user-defined run-modes that clock source may not be configured/enabled. - For S32K platform, PWM, ICU, and ADC drivers require configuration of the SIM register (which is in MCU). However, none of these drivers describe the required configuration. - For S32K platform, PWM, GPT, and ADC drivers require configuration of the trigger mux (which is in MCL). However, none of these drivers describe the required configuration.
MCAL-16352	New	New Feature [ICU] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures

Table continues on the next page...

ID	Subtype	Headline and Description
		for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-16354	New	New Feature [BASE] Rename Sys_GoToSypervisor into Sys_GoToSupervisor

Table continues on the next page...

ID	Subtype	Headline and Description
		Update MCAL_Arm.h to rename Sys_GoToSypervisor into Sys_GoToSupervisor
MCAL-16363	New	New Feature [PWM] Check validity during configuration time for reference parameters # Some parameters defined as references are mandatory, but it is not enforced by the xdm schema that they should not be empty, so if they are not configured Tresos issues errors only later, at generation step. This is due to missing validity check in the xdm file. For example, the WDG parameter WdgExternalTriggerCounterRef is mandatory. Currently Tresos issues the following error error: The XPath-expression "text:split(./WdgExternalTriggerCounterRef ,'/')[5]" caused an error: No value for xpath: text:split(./ WdgExternalTriggerCounterRef ,'/')[5] The proposed solution is to add a validity check in the xdm to ensure that mandatory reference parameters are not empty.In case of empty or invalid reference an appropriate error text should be output, such as: "Invalid or empty reference" Example of implementation: <a:da name="INVALID" type="XPath"> <a:tst expr="node:refvalid(.)" false="Invalid or empty reference."/> </a:da>
MCAL-16364	New	New Feature [PWM] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-16367	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[I2C] Update some defines in Reg_eSys_LPI2C.h and add support for LPI2C_1 NewWork Description: Some changes from new Reference Manual(Rev3) for SPI module: 1. LPI2C changes: - The "Table 46-1" in the subchapter "46.1.1 Instantiation information": Added the information for S32K142, S32K146, S32K148. - Change size of some fields: + MFCR[RXWATER]: from [23-16] to [17-16]. + MFCR[TXWATER]: from [7-0] to [1-0]. + MFSR[RXCOUNT]: from [23-16] to [18-16]. + MFSR[TXCOUNT]: from [7-0] to [2-0]. - Added new fields: SCR[RRF], SCR[RTF]. 2. FlexIO changes: - Added the subchapters: + 48.4.5 Interrupts and DMA Requests + 48.4.6 Peripheral Triggers Requirement source: - S32K14x reference manual: Rev.3, 03/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Need to update some defines in Reg_eSys_LPI2C.h: + LPI2C_MFCR_RXWATER_MASK_U32 + LPI2C_MFCR_TXWATER_MASK_U32 + LPI2C_MFSR_RXCOUNT_MASK_U32 + LPI2C_MFSR_TXCOUNT_MASK_U32 - Add support for LPI2C_1.
MCAL-16368	Bug	[BASE] Change QSPI base address to QSPI_0_BASEADDR<*>Detailed description (how to reproduce it): <*>Change QSPI Base Address to QSPI_0_BASEADDR instead of QSPI_BASEADDR <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>FLASH driver does not compile <*>Expected behavior: <*>Proposed solution (Optional): <*>See the description
MCAL-16370	New	New Feature [FLS] Analyze the impact of using the same resources by FLS/EEP/CSEC and update drivers Analyze the impacts of using the same resources by FLS/EEP/CSEC and update drivers. Consider: protection scheme, FCCOB, CCIF, and other hardware features which might influence the 3 drivers.
MCAL-16374	New	New Feature [MCL] Add FTM resources for S32K146 and S32K148 Detailed description (how to reproduce it): New S32K chips have added 4 FTM modules: FTM4,FTM5,FTM6,FTM7 Please support them in Ftm_Common.c and Ftm_Common_Types.h Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-16380	Bug	[ICU]Error template files<*>Detailed description (how to reproduce it): <*>In EBtreos,When I select s32k148_lqfp176 for Resource module and set FTM4 - FTM7 for Icu module <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>TS_Icu_017 <*>Observed behavior: <*>Error: <*>Parsing file C:\EB\2100\plugins\Icu_TS_T40D2M10I1R0/generate_PC/Icu_PluginMacros.m", line "106" The XPath-expression "num:i($ch)=$idx" caused an error: (2085) Cannot parse value "" to an int-value <*>Expected behavior: <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16388	Bug	[WDG] Warnings when Wdg is loaded in Tresos<*>Problem detailed description (how to reproduce it):<*>When Wdg module is loaded in Tresos, the following warnings are reported:<*>Ignoring attribute "UPPER-MULTIPLICITY" of "/TS_T2D47M10I0R0/Wdg/WdgSettingsConfig": Unknown attribute.<*>Ignoring attribute "LOWER-MULTIPLICITY" of "/TS_T2D47M10I0R0/Wdg/WdgSettingsConfig": Unknown attribute.<*>Issue comes from these 2 attributes defined for WdgSettingsConfig container in Wdg.xmd: <*><a:a name="LOWER-MULTIPLICITY" value="1"/><*><a:a name="UPPER-MULTIPLICITY" value="*"/><*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger:<*>[...] <*>Observed behavior:<*>[...] <*>Expected behavior:<*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>Remove LOWER-MULTIPLICITY and UPPER-MULTIPLICITY for WdgSettingsConfig node.
MCAL-16389	New	New Feature [ADC] Refactoring of ADC limit checking code - remove the correlated checks for SetChannel and limit checking, this is driver limitation - generate in the adc group configuration a field stating whether the group uses limit checking or not, to replace the current check via channel configuration that needs the unit id and first channel id and so on. - move in the generic code also the code to that handles the alignment, on platforms where that is available (Adcdig, CCs); the data mask (ADCDIG_CDR_DATA_MASK_U32) would have to be added as parameter for

Table continues on the next page...

ID	Subtype	Headline and Description
		generic function Adc_CheckConversionValuesInRange.
MCAL-16391	New	New Feature [Resource] Add ETH driver to RES_MODULES variable Add ETH driver to RES_MODULES variable in Resource.cfg file.
MCAL-16400	Bug	[ADC] Resolve the misra error for S32K<*>Detailed description (how to reproduce it): <*>Resolve the misra errors in the attachment <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix misra error
MCAL-16402	New	New Feature [OS] Create stub drivers for supporting OSCounterRef in Can driver In order to allow CAN driver to refer to OSCounter for getting timeout duration, it is necessary to have OS driver. We do not provide full implementation of OS, just basic configuration and function prototype for supporting CAN module.
MCAL-16404	Bug	[CAN] Can_Write() must not work in Can Listen Only Mode<*>Detailed description (how to reproduce it): <*>According S32K14x Series Reference Manual, Rev. 3, 03/2017, page 1553: <*>"This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters described in CAN_ECR register are frozen and the module operates in a CAN Error Passive mode". it means that we can not use Can_Write() in Listen only mode, if it is called it may return CAN_NOT_OK, or return some notify for user know what happen. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16405	Bug	[CAN] Duplicate code line<*>Detailed description (how to reproduce it): <*>This issue happen in Can_Flexcan.c, API: Can_FlexCan_SetControllerToStartMode: <*>code line 1576 and 3555 are duplicate because when calling Can_FlexCan_InitController() (1546), it will execute code line: 3555 and when finishing, code line:1576 also execute with the same instruction: REG_BIT_CLEAR32(FLEXCAN_MCR(u8HwOffset), FLEXCAN_MCR_SUPV_U32);. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16406	Bug	[CAN] CanAbortOnlyOneMB option must depend on CanApiEnableMbAbort<*>Detailed description (how to reproduce it): <*>In configuration time, driver must check CanApiEnableMbAbort option before permiting user use CanAbortOnlyOneMB option, it means: <*>CanApiEnableMbAbort = ON-> CanAbortOnlyOneMB=ON/OFF <*>CanApiEnableMbAbort = OFF-> CanAbortOnlyOneMB=OFF <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16407	New	New Feature [RTE] Update Mcal_TrustedCall Update Mcal_TrustedCall in the function calls
MCAL-16408	Bug	[CAN] Wrong logic code<*>Detailed description (how to reproduce it): <*>This issue happen in Can_Flexcan.c, "#if (CAN_RXFIFO_ENABLE == STD_ON)" (line: 2615, 2629) are unnecessary because this compiler condition was put in the begin of the Can_FlexCan_ProcessRxFifo function. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16409	Bug	[CAN] CanTableIDType must not accept Extended ID when CanExtendedIDSupport=OFF<*>Detailed description (how to reproduce it): <*>In current implementation for CAN driver, driver still accepts Extended frame ID using FIFO even CanExtendedIDSupport=OFF. Driver should check in generate time to prevent this issue. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16413	New	New Feature [MCU] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting

Table continues on the next page...

ID	Subtype	Headline and Description
		functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16416	New	New Feature [I2C] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16419	New	New Feature [DIO] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling

Table continues on the next page...

ID	Subtype	Headline and Description
		trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16420	New	New Feature [MCL] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16421	New	New Feature [FLS] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when

Table continues on the next page...

ID	Subtype	Headline and Description
		calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16422	New	New Feature [FEE] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16424	New	New Feature [CAN] Update Mcal_TrustedCall functions for user mode support User mode implementation in Base has been updated and the driver implementation should be updated accordingly: Mcal_Trusted_Call(name) shall be used when calling trusting functions with no parameter and no return Mcal_Trusted_Call1param(name,param) shall be used when calling trusting functions with 1 parameter and no return Mcal_Trusted_Call2params(name,param1,param2) shall be used when calling trusting functions with 2 parameters and no return Mcal_Trusted_Call3params(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and no return Mcal_Trusted_Call4params(name,param1,param2,param3,param4) shall be

Table continues on the next page...

ID	Subtype	Headline and Description
		used when calling trusting functions with 4 parameters and no return Mcal_Trusted_Call_Return(name) shall be used when calling trusting functions with no parameter and with return Mcal_Trusted_Call_Return1param(name,param) shall be used when calling trusting functions with 1 parameters and with return Mcal_Trusted_Call_Return2param(name,param1,param2) shall be used when calling trusting functions with 2 parameters and with return Mcal_Trusted_Call_Return3param(name,param1,param2,param3) shall be used when calling trusting functions with 3 parameters and with return Mcal_Trusted_Call_Return4param(name,param1,param2,param3,param4) shall be used when calling trusting functions with 4 parameters and with return
MCAL-16452	Bug	[ICU] Icu_Ftm_DeInit should not DeInit all FTM modules<*>Detailed description (how to reproduce it): <*>Icu_Ftm_DeInit should not DeInit all FTM modules, <*>Observed behavior: <*>In function Icu_Ftm_DeInit of Icu_Ftm.c: <*>{color:red} for (u8hwModuleNo = 0U ; u8hwModuleNo < ICU_NUM_FTM_HW_MODULE_U8 ; u8hwModuleNo++) <*>{ <*>Icu_Ftm_GlobalConfiguration((uint8)u8hwModuleNo, (uint8)0, (uint8)0); <*>}{color} <*>---> Icu_Ftm_DeInit will deinit all FTM modules <*>Expected behavior: <*>---> Icu_Ftm_DeIni only deinit FTM modules that are used by ICU <*>
MCAL-16457	Bug	[I2C] New errata e10792 implementation for I2C<*>Detailed description (how to reproduce it): <*>e10792: LPI2C: Slave Transmit Data Flag may incorrectly read as one when TXCFG is <*>zero. <*>Description: When SCFGR1[TXCFG] = 0, the slave transmit data ready flag can incorrectly assert for one cycle. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>e10792: When SCFGR1[TXCFG] = 0, the slave transmit data ready flag can incorrectly assert for one cycle. <*>Expected behavior: <*>Set SCFGR1[TXCFG] = 1 in Slave mode <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>In Slave mode: <*>- Set SCFGR1[TXCFG] = 1 <*>- Disable Transmit Data Interrupt in the function I2C_LPI2C_InitChannel(). <*>- Disable Transmit Data Interrupt when: <*>+ STOP bit is detected. <*>+ Slave has detected a Repeated START condition. <*>+ Slave has detected a error. <*>- Enable Transmit Data Interrupt when Slave has received ADDR0 matching address.
MCAL-16551	Bug	[PORT] Correct the range of PortPinPcr and PortPinId for S32K<*>Detailed description (how to reproduce it): <*>Some pins for S32K148 can not configure, the reason is the range of PortPinPcr andPortPinId for Kinetis is not updated when adding support for S32K146 and S32K148 <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Correct the range of PortPinPcr: from 0-144 to 0-155, PortPinId: from 0-145 to 0-156

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16552	New	New Feature [GPT] Implement workaround for sRTC e10716 Errata has trigger condition in the following code locations: - Gpt_SRtc_EnableInterrupt {{write to TAR register might be done when counter is enable}} - Gpt_SRtc_DisableInterrupt {{write to TAR register might be done when counter is enable}} - Gpt_SRtc_StopTimer {{write to TAR register might be done when counter is enable}} For all above cases the flag for time alarm must be checked and re-write TAR in case this was modified. The following 2 locations does not meet the trigger criteria and are considered safe: - Gpt_SRtc_ProcessCommonInterrupt - Gpt_SRtc_StartTimer *Once ticket is implemented please create BASE ticket to enable the errata in code.*
MCAL-16553	Bug	[CAN] Cannot write MB message when configure MB for each RAM block<*>Detailed description (how to reproduce it): <*>The driver jumps to hard fault handler when configured MB for each RAM block. <*>The problem is: <*>(uint16)0x0280U, /* Address of Message Buffer */ <*>(uint8)32U, /* Payload lenth of Message Buffer */ <*>(uint8)21U /* The index of MB in message buffer memory * <*>MB above does not exist in RAM block for this platform. <*>Preconditions: <*>+ configure at least one controller <*>+ CanSpecifiedRAMBlockSize is enabled <*>Test Case ID (internal TC that caught the defect) - optional <*>Can_TC_8006.c <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16573	Bug	[CAN] CanControllerRef field always invalid.<*>Detailed description (how to reproduce it): <*>At line 3298 of Can_42.xdm, appear the line check: <*>"_*<a:tst expr="node:refvalid(.) and (substring-before(substring-after(node:path(.),'/CanConfigSet/'),'/') != substring-before(substring-after(node:path(node:ref(.)),'/CanConfigSet/'),'/'))" true="You can not reference to out of configuration."/>*_" <*>This line is not necessary and it make the CanControllerRef always invalid on any case. <*>Preconditions: <*>run with ASR 422 <*>Test Case ID (internal TC that caught the defect) - optional <*>All test case will met this issue. <*>Observed behavior: <*>N/A <*>Expected behavior: <*>Remove the invalid states on all cases. <*>Proposed solution (Optional): <*>remove this line command.
MCAL-16575	Bug	[LIN][IPV_LPUART] Lin_GetStatus may return wrongly an error status in physical bus noise case<*>Detailed description (how to reproduce it): <*>1. When review the Lin's driver of IPV LPUART for S32K14X, i see that Lin_LPUART_HardwareGetStatus function (in Lin_LPUART.c file) has an incorrect condition as below: <*>- In transmission process of response of a frame, if Noise Error has occurred (NF flag is set: LPUART_STAT[NF] == 1)

Table continues on the next page...

ID	Subtype	Headline and Description
		then Lin's driver always returns state of the current transmission is LIN_RX_ERROR: <*><Source code> <*>/* Gets the status of the LIN driver when Channel is Header transmission */ <*>... <*>/* Gets the status of the LIN driver when Channel is Data transmission */ <*>case LIN_NOISE_ERROR: <*>... <*>u8TempReturn = LIN_RX_ERROR; <*>break; <*>... <*><Source code> <*>2. From my opinion, we need to distinguish that Noise Error occurred in process of transmit (Master response) or process of receipt (Slave response): <*><Source code> <*>/* Gets the status of the LIN driver when Channel is Header transmission */ <*>... <*>/* Gets the status of the LIN driver when Channel is Data transmission */ <*>case LIN_NOISE_ERROR: <*>if(LIN_TX_SLAVE_RES_COMMAND == u8TransmitHeaderCommand) <*>{ <*>u8TempReturn = LIN_RX_ERROR; <*>} <*>else <*>{ <*>u8TempReturn = LIN_TX_ERROR; <*>} <*>break; <*>... <*><Source code> <*>Preconditions: <*>Send a data frame with noise detected in physical bus (LPUART_STAT[NF] == 1). <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>LIN driver should be return status incorrectly in case noise occurred. <*>Expected behavior: <*>LIN driver should be return status correctly. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Modify driver as the above solution.
MCAL-16576	New	New Feature [BASE] Add errata for SRTC e10716: RTC: Timer Alarm Flag can assert erroneously Description: Writing to the Time Alarm Register (RTC_TAR) at the same time the RTC Seconds Register is incrementing can assert the Time Alarm Flag (RTC_SR[TAF]) bit in the RTC Status Register. Workaround: Write the Time Alarm Register (RTC_TAR) when the RTC Seconds Register is not incrementing. This can be when Time Counter Enable (RTC_SR[TCE]) bit in the RTC Status Register is clear or within the RTC_SR[TAF] interrupt routine. Alternatively, if the RTC_SR[TAF] is asserted following a write to the RTC_TAR, then write the RTC_TAR again.
MCAL-16579	Bug	[DIO] Fix the code after review against checklist for S32K14x RTM 1.0.1 ASR 4.0<*>Detailed description (how to reproduce it): <*>After implement ticket MCAL-15870: [DIO] Perform code review against checklist for S32K14x RTM 1.0.1 ASR 4.0. Some findings is found and need to be fixed <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the findings what is listed in the report (attached)
MCAL-16580	Bug	[PORT] Fix the code after review against checklist for S32K14x RTM 1.0.1 ASR 4.0<*>Detailed description (how to reproduce it): <*>After implement ticket

Table continues on the next page...

ID	Subtype	Headline and Description
		MCAL-15871: [PORT] Perform code review against checklist for S32K14x RTM 1.0.1 ASR 4.0. Some findings is found and need to be fixed <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the findings what is listed in the report (attached)
MCAL-16581	New	New Feature [BASE] Update MemMap header file for each driver to contain a section called .ac<driver_name>_code_rom Update MemMap header file for each driver to contain a section called .ac<driver_name>_code_rom. Right now, every driver contain a section called .acfls_code_rom. Instead of having this unique section for all drivers, every driver should contain it's own section called .ac<driver_name>_code_rom.
MCAL-16591	New	New Feature [ICU] Add implementation for PR-MCAL-3242.icu In Kinetis Cut 2.1 we have bit CHIS from FTM_CSC that allows to read the pin input level. This means that requirement PR-MCAL-3242.icu now also applies on Kinetis, as it was added by the ITWG team.
MCAL-16597	Bug	[PWM]Wrong duty and period when calling Pwm_SetDutyCycle_NoUpdate() or Pwm_SetPeriodAndDuty_NoUpdate() as soon as after Pwm_Init()<*>Detailed description (how to reproduce it): <*>-DutyCycle is updated when I call Pwm_SetDutyCycle_NoUpdate() as soon as after Pwm_Init().As expectation , Dutycycle is only updated when I call Pwm_SynceUpdate() function. <*>- When debugging, the bit SWSYNC is set to 1 when I call Pwm_SetDutyCycle_NoUpdate()after Pwm_Init() and it make the Duty updated. <*>- This issue is same for Pwm_SetPeriodandDutyCycle_Noupdate(). <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16601	New	New Feature [EEP]Provide API information to be used in generating bswmd file for Eep plugin Provide information that will be used in creating bswmd file for Eep plugin. APIs, Service ID, Sync/Async, and Reentrancy Example for Wdg: Module WDG Wdg_Init 0x00 sync non_reentrant Wdg_SetMode 0x01 sync non_reentrant Wdg_SetTriggerCondition 0x03 sync non_reentrant

Table continues on the next page...

ID	Subtype	Headline and Description
		Wdg_GetVersionInfo 0x04 sync reentrant Wdg_Cbk_GptNotification 0x05 sync non_reentrant
MCAL-16604	New	New Feature [EEP] Create EEP driver for S32K 4.2 release Create EEP driver for S32K release
MCAL-16608	New	New Feature [MCU] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16609	New	New Feature [LIN] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16610	New	New Feature [SPI] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16611	New	New Feature [I2C] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16612	New	New Feature [ADC] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16613	New	New Feature [DIO] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017)

Table continues on the next page...

ID	Subtype	Headline and Description
		- Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16614	New	New Feature [PORT] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16617	New	New Feature [ICU] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16618	New	New Feature [PWM] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team).

Table continues on the next page...

ID	Subtype	Headline and Description
		- Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16621	New	New Feature [WDG] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16623	New	New Feature [FEE] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16624	New	New Feature [CAN] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version>

Table continues on the next page...

ID	Subtype	Headline and Description
		Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16625	New	New Feature [ETH] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16628	New	New Feature [BASE] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)
MCAL-16629	New	New Feature [RESOURCE] Revision history for UM, IM should be generic NewWork Description: Remove revision_history.xml file from UM, IM folders. Instead, the generic content revision_history.xml will be used, it will be updated once per release by build_env owner, and not by each developer. The file file is added in build_env and will be automatically added in the archive to be converted by SSDS tool. The content should be as below: - Revision: 1.0 - Date: release date (ex: 24-Feb-2017) - Author: NXP MCAL team (because we have TS from test team and UM/IM from dev team). - Description: Version for <Platform> <autosar version> <Release version> Release. (ex: Version for S32K 4.0 RTM 1.0.0 Release)

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16631	New	New Feature [BASE] Remove errata ERR_IPV_FMC_0001 from Base Remove errata ERR_IPV_FMC_0001 from Base
MCAL-16634	Bug	[ADC] Fix code review against checklist findings for S32K14x RTM 1.0.1 ASR 4.0<*>Detailed description (how to reproduce it): <*>There are some findings in code review against checklist. See attach file for more detail. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix code review findings
MCAL-16647	Bug	[CRCU] Incorrect usage of Memory Allocation Sections<*>Detailed description (how to reproduce it): <*>In CDD_Crcu_PBCfg.c, the declaration of PB configuration data set will be generated under CONST_UNSPECIFIED section when calling Crcu_GenerateConfig macro under. <*>However, in CDD_Crcu_Cfg.h, it is externed under CONFIG_DATA_UNSPECIFIED section. <*>This make error at build by IAR compiler: the declaration in Crcu_PBcfg.c is "incompatible" with the one Crcu_cfg.h. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>This error will disappear. <*>Expected behavior: <*>This declaration is compatible in both files. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>In CDD_Crcu_PBCfg.c, calling Crcu_GenerateConfig macro under CONFIG_DATA_UNSPECIFIED section.
MCAL-16650	Bug	[I2C] Fix compiler warnings<*>Detailed description (how to reproduce it): <*>I2C driver has compiler warnings with the variable <*>"static VAR(I2C_FlexIO_HwChannelConfigType, I2C_VAR) I2C_FlexIOChannelConfig" in I2C_RegOperations.m file. <*>warning message: possible conflict for segment/section ".mcal_const_cfg": variable "I2C_FlexIOChannelConfigPB0Ch0 @ ".mcal_const_cfg"" (declared at line 190 of "e:/local_01/output/projects/ar_int_i2c_iar/target/build/I2C_TSe_Eqb_001_cfg1/generate/src/I2C_PBcfg.c") is an initialized variable variable "I2C_LPI2CChannelConfigPB0Ch1 @ ".mcal_const_cfg"" (declared at line 309 of "e:/local_01/output/projects/ar_int_i2c_iar/target/build/I2C_TSe_Eqb_001_cfg1/generate/src/I2C_PBcfg.c") is a constant (3 more variables like this) <*>Preconditions: <*>Generate compiler warning report <*>Test Case ID (internal TC that caught the defect) - optional <*>I2C_TSe_Eqb_001 <*>i2c_ts_0009 <*>tsi_bbx_fnc_i2c_001 <*>tsi_bbx_fnc_i2c_002 <*>Observed behavior: <*>Appear compiler warning <*>Expected behavior: <*>No compiler warning <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Replace: <*>static VAR(I2C_FlexIO_HwChannelConfigType, I2C_VAR) I2C_FlexIOChannelConfig <*>With: <*>static CONST(I2C_FlexIO_HwChannelConfigType, I2C_CONST)

Table continues on the next page...

ID	Subtype	Headline and Description
		I2C_FlexIOChannelConfig
MCAL-16651	Bug	[LIN] Fix Misra violation<*>Detailed description (how to reproduce it): <*>In the current code, there are some misra violations in the following file: Lin_LPUART.c. For detail, refer to attached misra log. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Misra violations were fixed or commented. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Please see the analysis tab
MCAL-16652	Bug	[I2C] Fix misra errors<*>Detailed description (how to reproduce it): <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1147, Violates [Encompasses MISRA 2004 Rule 11.1, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1147, Violates [MISRA 2004 Rule 11.3, advisory]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1151, Violates [Encompasses MISRA 2004 Rule 11.1, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1151, Violates [MISRA 2004 Rule 11.3, advisory]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1296, Violates [Encompasses MISRA 2004 Rule 11.1, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1296, Violates [MISRA 2004 Rule 11.3, advisory]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1300, Violates [Encompasses MISRA 2004 Rule 11.1, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_FlexIO.c", line 1300, Violates [MISRA 2004 Rule 11.3, advisory]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_LPI2C.c", line 401, Violates [MISRA 2004 Rule 1.4, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins\I2c_TS_T40D2M10I1R0\src\I2C_LPI2C.c", line 401, Violates [MISRA 2004 Rule 1.4, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb

Table continues on the next page...

ID	Subtype	Headline and Description
		\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \I2c_TS_T40D2M10I1R0\src\I2C_LPI2C.c", line 1112, Violates [MISRA 2004 Rule 1.4, required]: no MISRA violation comment was found (maybe wrong format is used). <*>File "c:\vv_tools\eb \EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \I2c_TS_T40D2M10I1R0\src\I2C_LPI2C.c", line 1113, Violates [MISRA 2004 Rule 1.4, required]: no MISRA violation comment was found (maybe wrong format is used). <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-16658	Bug	[MCU] Fix misra errors and compiler warnings S32K14X<*>Detailed description (how to reproduce it): <*>Mcu driver violate some misra rules and compiler warnings <*>Preconditions: <*>Any <*>Test Case ID (internal TC that caught the defect) - optional <*>Any <*>Observed behavior: <*>Mcu driver violate some misra rules <*>Expected behavior: <*>Mcu driver will not violate some misra rules <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix misra errors and compiler warnings <*>
MCAL-16660	Bug	[ICU] Fix misra errors S32K14X<*>Detailed description (how to reproduce it): <*>Icu driver violate some misra rules <*>Preconditions: <*>Any <*>Test Case ID (internal TC that caught the defect) - optional <*>Any <*>Observed behavior: <*>Icu driver violate some misra rules <*>Expected behavior: <*>Icu driver will not violate some misra rules <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix misra errors
MCAL-16675	Bug	[I2C] Some files do not have an empty line at the end<*>Detailed description (how to reproduce it): <*>The following files do not have an empty line at the end: <*>I2C_CFG_42.h <*>I2C_RegOperations.m <*>I2C_RegOperations_42.m <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>The empty line does not exist <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Insert an empty line
MCAL-16682	Bug	[CAN]Fix error of compiler warning<*>Detailed description (how to reproduce it): <*>In the compiler warning report which contains some warnings <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>when run compiler warning the report haven't warning. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16683	Bug	[ICU] Checking resource is not correct when generating plugin<*>Detailed description (how to reproduce it): <*>ERROR 17-05-08,15:38:10 (2004) [D:\Installed\EB\2100\plugins\Icu_TS_T40D2M10I1R0/generate_PB/Icu_PluginMacros.m:323]: PORT_4 do not contain channel index greater than 16, refer resource to set correctly (node) <*>ERROR 17-05-08,15:38:10 (2070) Failed to generate file "D:\Installed\EB\2100\plugins\Icu_TS_T40D2M10I1R0/generate_PB/src/Icu_PBcfg.c" <*>Error "Duplicate physical channel" is not clear <*>Preconditions: <*>edit configset 121 from PORT4_CH_17 to PORT4_CH_19 with resource s32k148_lqfp144 <*>Test Case ID (internal TC that caught the defect) - optional <*>TEST=Icu_TS_017, with configset 121, PORT4_CH_19 <*>Observed behavior: <*>PORT_4 do not contain channel index greater than 16, refer resource to set correctly (node) <*>Error "Duplicate physical channel" is not clear <*>Expected behavior: <*>Checking resource correctly <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Checking resource correctly in template and config files, making clearly "Duplicate physical channel"
MCAL-16686	Bug	[CAN] Missing a limitatiton when using do...while loop<*>Detailed description (how to reproduce it): <*>This issue happen in Can_Flexcan.c (line 2517 - 2738, function: Can_FlexCan_ProcessRxFifo), driver using do...while loop, but it did not use a limitation to exit the loop, this may take driver falls in infinite loop in unexpected condition. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16687	Bug	[ICU] Fix the code review against checklist finding for S32K<*>NewWork Description: <*>- Review code against checklist. <*>- Fill the review result to the checklist. <*>- The checklist template and coding guideline are enclosed in attachment. <*>Requirement source: <*>sMCAL Release criteria document version 5.1: <*>http://compass.freescale.net/go/228798570 <*>Coding guideline verion 5.0 date 19-Jul-2016: https://www.nxp.com/go/230979668 <*>Code review checklist verion 4.0, date Nov-2016: https://www.nxp.com/go/230979668 <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): <*>- The result of this activity will be the code review checklist and this ticket is treated as ?platform specific?. <*>- Code Review checklists, both Intermediate and Final, should be added in GIT and attached to this ticket. <*>- New ticket will be raised for the code modifications resulted for the ?code review? activity. This ticket will be used to update code, will be analyzed for all platforms and changes will be integrated for all affected platforms. <*>
MCAL-16688	Bug	[I2C] The driver cannot clear interrupt flags when the driver has not been initialized<*>Detailed description (how to reproduce it): <*>- The driver cannot clear interrupt flags when the driver has not been initialized. <*>- In Slave

Table continues on the next page...

ID	Subtype	Headline and Description
		mode, the driver cannot clear interrupt flags when the errors is present. So, the driver always generate interruption. <*>Preconditions: <*>Run the test suite: tse_bbx_fnc_i2c_009 <*>Test Case ID (internal TC that caught the defect) - optional <*>tse_bbx_fnc_i2c_009 <*>Observed behavior: <*>interruption always occurs when the driver has not been initialized. <*>Expected behavior: <*>Reset all register if the driver has not been initialized <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>- Reset all register if the driver has not been initialized by interrupt functions I2C_FlexIO_InterruptHandler, I2C_LPI2C_MasterInterruptHandler, I2C_LPI2C_SlaveInterruptHandler <*>- In slave mode, clear interrupt flags when the errors is present.
MCAL-16690	New	New Feature [CAN] Missing clear FIFO contents operation when driver go to stop mode Detailed description (how to reproduce it): According requirement CAN282: "The function Can_SetControllerMode(CAN_T_STOP) shall cancel pending messages", driver is still missing when doing cancel pending messages without cancel FIFO operation. (see function: Can_FlexCan_SetControllerToStopMode, line: 1405) Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-16696	Bug	[CAN] A code paragraph in Can_FlexCan_ProcessRxFifo can't be implemented<*>Detailed description (how to reproduce it): <*>In Can_FlexCan_ProcessRxFifo function has a code paragraph can't be implemented because this function was implemented only Rx FIFO was enabled but the following code only was implemented when ((CAN_CONTROLLERCONFIG_RXFIFO_U32 != ((Can_pControlerDescriptors[controller].u32Options) & CAN_CONTROLLERCONFIG_RXFIFO_U32)) || (Can_pControlerDescriptors[controller].u8RxFiFoUsedMb <= (uint8)u8MbIndex). This happen when isn't in Rx Fifo. <*>The detail about code paragraph: <*>#if (CAN_RXFIFO_ENABLE == STD_ON) <*>/* @violates @ref Can_Flexcan_c_REF_6 Violates MISRA 2004 Required Rule 17.4, pointer arithmetic other than array indexing used */ <*>if ((CAN_CONTROLLERCONFIG_RXFIFO_U32 != ((Can_pControlerDescriptors[controller].u32Options) & CAN_CONTROLLERCONFIG_RXFIFO_U32)) || <*>/* @violates @ref Can_Flexcan_c_REF_6 Violates MISRA 2004 Required Rule 17.4, pointer arithmetic other than array indexing used */

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>(Can_pControlerDescriptors[controller].u8RxFiFoUsedMb <= (uint8)u8MbIndex) <*>) <*>{ <*>#endif <*>if(FLEXCAN_MBCS_CODERXOVRR_U32 == (u32MbConfig & FLEXCAN_MBCS_CODE_U32)) <*>{ <*>/* bMbOver = (VAR(boolean, AUTOMATIC))TRUE; */ <*>(void)Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_READ, (uint8)CAN_E_DATALOST); <*>} <*>#if (CAN_RXFIFO_ENABLE == STD_ON) <*>} <*>#endif /* (CAN_RXFIFO_ENABLE == STD_ON) */ <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Delete the above code paragraph.
MCAL-16699	Bug	[CAN] Incorrect rounding number for time duration<*>Driver need to do division of float parameter to calculate the number of ticks need to wait during timeout. <*>Can_u32TicksDelayDuration = (uint32)(CAN_TIMEOUT_DURATION / CAN_TIMEOUT_COUNTER_TICK2NS()); <*>This function rounding is not properly, it could lead to the Delay duration shorter than expected. <*>In order to make driver work more stable, the TickDelayDuration should be ceiling of this equation. <*>
MCAL-16703	Bug	[CAN] Mcal_Trusted_Call_Return is not work when using with OS<*>Detailed description (how to reproduce it): <*>In can driver, the Mcal_Trusted_Call_Return is implemented as following: <*>#define Can_IPW_InitController(Controller) Mcal_Trusted_Call_Return1param((uint32)Can_FlexCan_InitController, (Controller)) <*>This is not compatible with definition when OS is using: <*>#define Mcal_Trusted_Call_Return(name) Call_##name##_TRUSTED() <*>This might lead to build fail when OS is used. <*>Preconditions: <*>-OS is used <*>-User mode is enable <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Build fail <*>Expected behavior: <*>build OK <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Correct the define to compatible with BASE
MCAL-16706	Bug	[ADC] Build fail with case : Adc Optimize OneShot HwTrigger Conversions and ADC_ENABLE_LIMIT_CHECK ON<*>Detailed description (how to reproduce it): <*>In Adc_Adc12bsarv2_EndDmaPartialConversion function: <*>In case of Optimize OneShot HwTrigger and limit checking use together, there is an "else" statement without "if" condition. <*>the code like that: <*>#if (ADC_OPTIMIZE_ONESHOT_HW_TRIGGER == STD_OFF) <*>#if (ADC_ENABLE_LIMIT_CHECK == STD_ON) <*>if(<condition>) <*>#endif <*>#endif <*>/*...*/ <*>#if (ADC_ENABLE_LIMIT_CHECK == STD_ON) <*>else <*>{ <*>/* ... */ <*>} <*>#endif <*>so it raise error at build when ADC_OPTIMIZE_ONESHOT_HW_TRIGGER == STD_ON and ADC_ENABLE_LIMIT_CHECK == STD_ON <*>=> add the pre-compile condition for "else" statement under (ADC_OPTIMIZE_ONESHOT_HW_TRIGGER == STD_OFF)

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>Fix the fail at build in that case <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>=> add the pre-compile condition for "else" statement under (ADC_OPTIMIZE_ONESHOT_HW_TRIGGER == STD_OFF)
MCAL-16709	Bug	[MCL] Build error on S32K14X with IAR compiler<*>Detailed description (how to reproduce it): <*>The MCL driver failed at build step on IAR compiler with below log: <*>"c:\vv_tools\eb \EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \Mcl_TS_T40D2M10I1R0\src\Mcl_Lmem.c",1549 Error[Pe513]: a value of type "void" cannot be assigned to an entity of type "uint8" <*>retVal = LMEM_CALL_CACHE_INVALIDATE_MULTILINES_PC(u32PhyAddr, u32Length); <*>^ <*>"c:\vv_tools\eb \EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \Mcl_TS_T40D2M10I1R0\src\Mcl_Lmem.c",1585 Error[Pe513]: a value of type "void" cannot be assigned to an entity of type "uint8" <*>retVal = LMEM_CALL_CACHE_INVALIDATE_MULTILINES_PC(u32PhyAddr, u32Length); <*>^ <*>"c:\vv_tools\eb \EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \Mcl_TS_T40D2M10I1R0\src\Mcl_Lmem.c",1593 Error[Pe513]: a value of type "void" cannot be assigned to an entity of type "uint8" <*>retVal = LMEM_CALL_CACHE_CLEAR_MULTILINES_PC(u32PhyAddr, u32Length); <*>^ <*>"c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \Mcl_TS_T40D2M10I1R0\src\Mcl_Lmem.c",1629 Error[Pe513]: a value of type "void" cannot be assigned to an entity of type "uint8" <*>retVal = LMEM_CALL_CACHE_CLEAR_MULTILINES_PC(u32PhyAddr, u32Length); <*>^ <*>"c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_00\plugins \Mcl_TS_T40D2M10I1R0\src\Mcl_Lmem.c",1637 Error[Pe513]: a value of type "void" cannot be assigned to an entity of type "uint8" <*>Issue is that in the macro LMEM_CALL_CACHE_CLEAR_MULTILINES_PC uses the macro Mcal_Trusted_Call2params for a non-void return function, the Mcal_Trusted_Call_Return2param function should be used instead. <*>Preconditions: <*>Build MCL driver with IAR compiler <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-16754	New	New Feature [ADC] Support continuous group without interrupt Currently the continuous group is not support without interrupt. We can support this feature by the way using PDB continuous mode.
MCAL-16755	Bug	[PWM] Fix findings after Perform code review against checklist for S32K14x RTM 1.0.1 ASR 4.0<*>NewWork Description: <*>Fix findings in the attached file <*>(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) <*>Proposed solution (Optional): NA <*>

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16756	Bug	[ADC] Dma shouldn't be re-configured at the middle of transfer<*>Detailed description (how to reproduce it): <*>- In the Adc_Pdb_ChannelSequenceError function, the calling Adc_Pdb_ReConfigureDma can be occurred at the middle of transfering. Dma configuration wil be not updated. <*>So <*>- In the Adc_Pdb_ChannelSequenceError function, the calling Adc_Pdb_ReConfigureDma should be moved before the clearing of PDB errors code. <*>- add a while loop with polling and timeout to ensure that DMA finished the transferring before re-configuring <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16757	New	New Feature [ADC] Update driver limitation for continuous group without interrupt Due to driver does not support for continuous group without interrupt. it should be updated in driver limitation chapter. One condition should be placed in xdm file to prevent this case.
MCAL-16765	Bug	[MCU] RAM Section Base Address and RAM Section Size are incorrect (S32K14X)<*>Detailed description (how to reproduce it): <*>RAM Section Base Address and RAM Section Size are incorrect (S32K14X) <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>RAM Section Base Address and RAM Section Size are incorrect (S32K14X) <*>Expected behavior: <*>RAM Section Base Address and RAM Section Size are correct (S32K14X) <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Correct RAM Section Base Address and RAM Section Size
MCAL-16767	Bug	[CAN] Spurious interrupts are not handled correctly in CAN driver<*>Detailed description (how to reproduce it): <*>Driver is trying to access to a NULL pointer (CanStatic_pCurrentConfig) in case spurious occur. The below code should check for NULL pointer before process. <*>if (CAN_STARTED != Can_ControllerStatuses[CAN_FC##FC##_INDEX].ControllerState) \ <*>{ \ <*>REG_WRITE32(FLEXCAN_ESR(CanStatic_pCurrentConfig->StaticControlerDescriptors[CAN_FC##FC##_INDEX].u8ControllerOffset), (FLEXCAN_ESR_ERRINT_U32 & FLEXCAN_ESR_CONFIG_MASK_U32));\ <*>} \ <*>else \ <*>{ \ <*>can_status = (uint32)REG_READ32(FLEXCAN_CTRL(FLEXCAN_##FC##_OFFSET)) & FLEXCAN_CTRL_ERRMSK_U32; \ <*>can_status = can_status >> (13U); \ <*>can_status &= ((uint32)REG_READ32(FLEXCAN_ESR(FLEXCAN_##FC##_OFFSET)) & FLEXCAN_ESR_ERRINT_U32); \ <*>if (0U != can_status) \ <*>{ \ <*>if (NULL_PTR != CanStatic_pCurrentConfig->StaticControlerDescriptors[CAN_FC##FC##_INDEX].Can_ErrorNotification) { \ <*>CanStatic_pCurrentConfig-

Table continues on the next page...

ID	Subtype	Headline and Description
		>StaticControlerDescriptors[CAN_FC##FC##_INDEX].Can_ErrorNotification(); \ <*>} \ <*>} \ <*>REG_WRITE32(FLEXCAN_ESR(CanStatic_pCurrentConfig->StaticControlerDescriptors[CAN_FC##FC##_INDEX].u8ControllerOffset), (FLEXCAN_ESR_ERRINT_U32 & FLEXCAN_ESR_CONFIG_MASK_U32));\ <*>} \ <*>EXIT_INTERRUPT(); \ <*>} <*>Preconditions: <*>None. <*>Test Case ID (internal TC that caught the defect) - optional <*>All external white box tests <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16770	Bug	[DIO] Fix misra for S32K14X<*>Detailed description (how to reproduce it): <*>Some misra error is reported (see attached file for more detail) <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the misra errors in attached file
MCAL-16771	Bug	[PORT] Fix misra for S32K14X<*>Detailed description (how to reproduce it): <*>Some misra error is reported (see attached file for more detail) <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the misra errors in attached file
MCAL-16772	Bug	[GPT] Channel status change shall be made before to start the hardware<*>Problem detailed description (how to reproduce it): <*>There is comment in Gpt_StartTimer: <*>/* Change GPT channel status.Channel status change shall be made before to start <*>the hardware in order to not change the channel status from EXPIRED to RUNNING*/ <*>However, the code is implemented in different way. <*>The hardware timer is started before channel status change. <*>If the Gpt_StartTimer is preempted by high priority tasks/ interrupts, and the hardware timer exipres before the channel status change to GPT_STATUS_RUNNING inside Gpt_StartTimer, the Gpt channel interrupt will occur, the GPT interrupt routine will stop channel, and set channel status to GPT_STATUS_EXPIRED. <*>After returned from the Gpt channel interrupt routine, the channel status is set to GPT_STATUS_RUNNING. <*>Preconditions: <*>- GPT channel is in ONESHOT mode <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>- After calling Gpt_StartTimer, the Gpt channel status is GPT_STATUS_RUNNING but the hardware timer is stopped. <*>Expected behavior: <*>- After calling Gpt_StartTimer, the Gpt channel status is GPT_STATUS_RUNNING and the hardware timer is running or both expired (stopped) <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-16783	Bug	[ADC] Incorrect DMA configuration might occur in PDB error interrupt for some types of groups<*>Detailed description (how to reproduce it): <*>- In the Adc_Pdb_ChannelSequenceError function, the calling Adc_Pdb_ReConfigureDma can occurr at the middle of a DMA transfer. Dma configuration will be not updated correctly and the group conversion will not have correct data. <*>- Although the driver performs polling to ensure dma transferred then re-configure the setting, Dma will be started and paused and restarted for each channel, so it's still possible that an error state in DMA will occur. <*>- This problem only occurs on 148 derivative, ghs compiler. <*>- It can be reproduced by Adc_TS_072 with the configuration added in the attachment <*>Preconditions: <*>Group using Delays, not using BackToBack mode; <*>Group using DMA <*>Delays / triggers configured in such a way that PDB errors might occur. <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>incorrect DMA configuration after PDB error interrupt, incorrect / incomplete ADC conversion results <*>Expected behavior: <*>no incorrect reconfiguration of DMA & PDB error interrupt should completely recover ADC from the error state <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Clear all COCO flags in PDB error interrupt, if the group does not use BackToBack mode (driver update) <*>
MCAL-16785	Bug	[MCU] Wrong clearing for Low Voltage Detect Flag<*>Detailed description (how to reproduce it): <*>In the Mcu_PMC_LowVoltage_ISR, the driver is trying to write 1 to LVDF bit to clear the flag, but in face, to clear the LVDF flag, driver must write 1 to LVDACK bit. <*>Preconditions: <*>Low Voltage Detect interrupt is enabled and Low Voltage Detect interrupt is occurred. <*>Test Case ID (internal TC that caught the defect) - optional <*>TBC <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A
MCAL-16787	Bug	[CAN] The interrupt service haven't handle full MB for S32K148 platform<*>Detailed description (how to reproduce it): <*>In RM we have full 32 MB for 3 controllers: A, B and C so In the driver (Can_Irq.c) we can see that it haven't support for FlexCAN_B, FlexCAN_C with full MB on platform S32K148 derivative. <*>We need to add more interrupt routine for Controller B and Controller C as below: <*>ISR(Can_IsrFCB_MB_16_31); <*>ISR(Can_IsrFCC_MB_16_31); <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16788	Bug	[CAN] The driver has implemented wrong for fifo interrupt.<*>Detailed description (how to reproduce it): <*>The Can driver jumps to handle hard fault when I configure for controller as below: <*>- Enabled FIFO feature <*>- Rx, Tx are interrupt mode. <*>- Number of MB less than 15. <*>Root cause: <*>Because in the driver was implementing wrong for

Table continues on the next page...

ID	Subtype	Headline and Description
		Can_FlexCan_ProcessTx function as below: <*>for (u8MbIndex = mbindex_start; u8MbIndex <= mbindex_end; u8MbIndex++) <*>{ <*>u16MbGlobalIndex = Can_ControllerStatuses[controller].u16MBMapping[u8MbIndex]; <*>if (CAN_TRANSMIT == (Can_pCurrentConfig->MBConfigContainer.pMessageBufferConfigsPtr[u16MbGlobalIndex].eMBType)) <*>{ <*>Can_FlexCan_ProcessTxPoll(controller, u16MbGlobalIndex); <*>} <*>} <*>we always have mbindex_end equal by 15 and If user configured Mbindex < 15 then the array Can_ControllerStatuses[controller].u16MBMapping[u8MbIndex] will not exist for some elements. It causes hard fault on chip. <*>Preconditions: <*>- Enabled FIFO feature <*>- Rx, Tx are interrupt mode. <*>- Number of MB less than 15. <*>Test Case ID (internal TC that caught the defect) - optional <*>Can_TC_1101 <*>tc_flt_can_10103.c <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16789	Bug	[OS] Correct malform UUID<*>Detailed description (how to reproduce it): <*>The UUIDs get from OS packages are detected as malformed in our environment. Correct to prevent error reported. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Use the correct form of UUIDs <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Correct the UUIDs form
MCAL-16793	Bug	[ADC] Wrong number formatting in epd files<*>Detailed description (how to reproduce it): <*>There is a wrong number formatting in epd files in AdcChannelDelay node as below: <*><LOWER-MULTIPLICITY>0.0</LOWER-MULTIPLICITY> <*><UPPER-MULTIPLICITY>0.0</UPPER-MULTIPLICITY> <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>Adc_TS_E01 <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Please see the fixes in attachment
MCAL-16803	New	New Feature [MCU] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is

Table continues on the next page...

ID	Subtype	Headline and Description

		SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size);
		instead of
		SET_USER_ACCESS_ALLOWED(baseAddr);
		Depending of the IP version this will write at the correct offset for all known versions of register protection.

MCAL-16806	New	New Feature [I2C] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16810	New	New Feature [MCL] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16811	New	New Feature [FLS] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400

Table continues on the next page...

ID	Subtype	Headline and Description
		For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16812	New	New Feature [FEE] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16814	New	New Feature [CAN] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16818	New	New Feature [GPT] Register protection handling should support also IP version 01.02.04 On REG_PROT version 01.02.04 the AREA 4 memory space(configuration bits - GCR) is situated at an offset that is dependent of the module register space

Table continues on the next page...

ID	Subtype	Headline and Description
		that is protected. For protection size 1KB - GCR is at offset 0x00900 For protection size 2KB - GCR is at offset 0x01200 For protection size 4KB - GCR is at offset 0x02400 For protection size 8KB - GCR is at offset 0x04800 For protection size 16KB - GCR is at offset 0x09000 For protection size 32KB - GCR is at offset 0x12000 The macro to be used in code is SET_USER_ACCESS_ALLOWED(baseAddr,prot_mem_size); instead of SET_USER_ACCESS_ALLOWED(baseAddr); Depending of the IP version this will write at the correct offset for all known versions of register protection.
MCAL-16826	Bug	[CAN] Wrong implementation for Can_FlexCan_EnableInterrupts function<*>Detailed description (how to reproduce it): <*>The problem occurs that the IMASK1 register did not enabled for controller B, C which it supports 16MBs only. <*>root cause: <*>In the function Can_FlexCan_EnableInterrupts() was implemented wrong the condition code as below: <*>for (u8RegCount=(uint8)0U; \ <*>u8RegCount < (CanStatic_pControlerDescriptors[u8Controller].u8NumberOfMB >> FLEXCAN_MB_SHIFT5BIT_U8); \ <*>u8RegCount++ \ <*>) <*>{ <*>REG_WRITE32(Can_IflagImask[u8RegCount] [u8HwOffset].u32CanImask, Can_ControllerStatuses[u8Controller].u32MBInterruptMask[u8RegCount]); <*>} <*>For those controller supports 16MB. We always have the array: u8RegCount < (CanStatic_pControlerDescriptors[u8Controller].u8NumberOfMB = 16 so when the condition shifts left 5 (SHIFT5BIT) then the result equal by 0. This is wrong with For {} loop. <*>Preconditions: <*>- Configure at least one controller B or C <*>- Use interrupt mode <*>Test Case ID (internal TC that caught the defect) - optional <*>Can_TC_0090.c <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16822	Bug	[FLS] Fix VSMD warning and error for unused SPI parameter<*>Detailed description (how to reproduce it): <*>The "<v:lst name="FlsSpiReference">" parameter from Fls.xdm has to have a minimum value of 1 for the multiplicity limit. <*>The parameter is not used. <*>Observed behavior: <*>Warning issued in VSMD check. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Modify the MIN limit to value 1.
MCAL-16840	Bug	[CAN] Implementation of Can_MainFunction_Mode is not correct<*>Problem detailed description (how to reproduce it):<*>According to [CAN431]: If these main functions are called from the BSW Scheduler and the Can module is not initialized, then it shall return immediately without performing any functionality and without raising a production error.<*>This means that all calls of Can_MainFunction_Mode should always do nothing if CAN is not initialized. In case of DET is OFF, in the first call of Can_MainFunction_Mode it will call Can_FlexCan_MainFunctionMode even though CAN is not initialized. The

Table continues on the next page...

ID	Subtype	Headline and Description
		check if Can module is not initialized should be always performed, not just when DET is ON.<*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger:<*>[...] <*>Observed behavior:<*>Can_MainFunction_Mode still performs functionality even though Can module is not initialized. <*>Expected behavior:<*>Can_MainFunction_Mode should returns immediately without performing any <*>functionality and without raising a production error when Can module is not initialized. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>FUNC(void, CAN_CODE) Can_MainFunction_Mode(void)<*>{<*>if (CAN_UNINIT == Can_eDriverStatus)<*>{ <*>#if (CAN_DEV_ERROR_DETECT == STD_ON) <*>(void)Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_MODE, (uint8)CAN_E_UNINIT); <*>#endif /* (CAN_DEV_ERROR_DETECT == STD_ON) */ <*>}<*>else<*>{<*>Can_IPW_MainFunctionMode(); <*>} <*>}
MCAL-16884	Bug	[LIN] Correct VSMD error<*>Detailed description (how to reproduce it): <*>There is an error when running VSMD report. <*>This error should be corrected. <*>Message: An ENUMERATION-PARAM-DEF must define at least one ENUMERATION-LITERAL-DEF. <*>Node: /TS_T40D2M10I1R0/Lin/LinGlobalConfig/LinChannel/LinHwChannel <*>Preconditions: <*>Running VSMD report <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>No error <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Please see the analysis.
MCAL-16912	Bug	[FLS] Follow up : Fix compiler warnings <*>Detailed description (how to reproduce it): <*>There are remaining compiler warnings need to be fix in the S32K14X RTM_1.0.1 release. <*>Those warnings are analysed and commented in the compiler warnings report. <*>Please see the attachment for more details. <*>Preconditions: <*>Check compiler warning in driver <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>There are compiler warning not fixed or commented in code <*>Expected behavior: <*>Fix those compiler warnings <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the remaining compiler warning in the report for next release of S32K14X AUTOSAR 4.0 MCAL RTM_1.0.1 release
MCAL-16916	Bug	[FLS] Job processing might not be successfully finished for QSPI sectors used in interrupt mode<*>Detailed description (how to reproduce it): <*>In IRQ mode, flash erase and write jobs on QSPI sectors could incorrectly be marked as failed and exited on timeout condition, after successful completion, if the Fls_MainFunction`s task is interrupted in a specific point of the driver code. <*>The issue could affect only jobs over QSPI sectors configured in IRQ mode. <*>There is no data corruption, the jobs complete successfully in hardware. <*>The IRQ mode is not the recommended operation mode of the flash driver, nor the most performant and it was not available and used in any previous

Table continues on the next page...

ID	Subtype	Headline and Description
		versions of the FLS driver. Thus, the functional impact of this issue can be considered to be lower and does not affect any major features of the driver. <*>The issue does not manifest itself if the first iteration of Fls_Qspi_SectorErase/Fls_Qspi_SectorWrite is exited before the entire erase/ write job completes in hardware. <*>Preconditions: <*>-QSPI sector used <*>-FLS_USE_INTERRUPTS is STD_ON <*>-difficult to reproduce, we only managed to reproduce it with the code instrumented for code coverage tests. we believe that given the extra instrumentation code added to the interrupt routine, which takes too long to execute, the interrupt routine never exits completely before the next interrupt fires, tail-chaining, so that not a single line of code is executed until the entire interrupt state machine execution ends. <*>Test Case ID (internal TC that caught the defect) - optional <*>Fls_TS_00401_Combine, config 2 <*>Observed behavior: <*>FLS job processing is not marked as successfully finished, always stays in PENDING or exists on timeout depending on configuration <*>Proposed solution (Optional): <*>- In functions which schedule interrupts(Fls_Write, Fls_Erase) ensure that the job pending status is set before enabling the interrupt source.
MCAL-16917	New	New Feature [FLS] Add missing label to support fault injection tests There are some labels for fault injection tests are not supported by FLS driver, we should consider to have them supported on driver to have all fault injection tests can be executed. Please see attachment for more details
MCAL-16954	Bug	[PORT] Add default configuration to list of signed files in plugin<*>Detailed description (how to reproduce it): <*>Add default configuration to list of signed files in plugin <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-16970	Bug	[ICU] Exclusive Area in ICU Integration manual Rev. 1.4 is not completely described<*>Imported from CQ ticket ENGR00390175.<*>Problem detailed description (how to reproduce it):<*>Baseline: <*>Incorrect list of Exclusive Area described in ICU Integration Manual. To be more specific: <*>Missing EAs:<*>ICU_EXCLUSIVE_AREA_08<*>ICU_EXCLUSIVE_AREA_10<*>ICU_EXCLUSIVE_AREA_11<*>ICU_EXCLUSIVE_AREA_12<*>ICU_EXCLUSIVE_AREA_16<*>ICU_EXCLUSIVE_AREA_17<*>ICU_EXCLUSIVE_AREA_18<*>ICU_EXCLUSIVE_AREA_19<*>ICU_EXCLUSIVE_AREA_20<*>ICU_EXCLUSIVE_AREA_21<*>ICU_EXCLUSIVE_AREA_26<*>ICU_EXCLUSIVE_AREA_27<*>ICU_EXCLUSIVE_AREA_28<*>Unavailable EAs:<*>ICU_EXCLUSIVE_AREA_23<*>ICU_EXCLUSIVE_AREA_24<*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger:<*>[...] <*>Observed behavior:<*>Some Exclusive Areas are missing but some others are not available. <*>Expected behavior:<*>Update Exclusive Areas in Integration Manual for ICU.<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		<*>[...]
MCAL-16978	Bug	[WDG] Fixing wrong definition of WdgMaxNumberOfInstances<*>Detailed description (how to reproduce it): <*>Build failed due to initilize an array bigger than fixed size <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>WdgMaxNumberOfInstances should be 5
MCAL-16983	Bug	[ICU] Fix disabled container in file config EB<*>Detailed description (how to reproduce it): <*>Some containers were disabled when open EB to config. Such as: <*>IcuSignalEdgeDetection, IcuSignalMeasurement, etc <*>==> Cannot choose options in these containers <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Some containers were disabled when open EB to config. Such as: <*>IcuSignalEdgeDetection, IcuSignalMeasurement, etc <*>more info in attached picture <*>Expected behavior: <*>can be chosen option in these containers <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Transform <a:da name="EDITABLE" value="false"/> to <a:da name="ENABLE" value="false"/>
MCAL-17023	New	New Feature [OCU] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below)

Table continues on the next page...

ID	Subtype	Headline and Description
		(if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-17025	New	New Feature [MCL] Add support for CHIS bit from the FTM_CSC register FTM version 5.00.04.00 (on Kinetis) supports CHOV and CHIS bitfields to read the output and the input of the FTM channel. The CHIS bit is not defined in Reg_eSys_Ftm.h, so the appropriate defines should be added.
MCAL-17036	Bug	[ADC] Follow up to fix code review findings<*>Problem detailed description (how to reproduce it): <*>Some violations of are found via CR MCAL-16559 (See the attachment for more detail) <*>Preconditions: <*>N/A <*>Test Case

Table continues on the next page...

ID	Subtype	Headline and Description
		ID (internal TC that caught the defect) - optional <*>N/A <*>Trigger: <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>Fix code review fidings <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix the findings which are listed in the attachment
MCAL-17046	New	New Feature [GPT] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else

Table continues on the next page...

ID	Subtype	Headline and Description
		#define Call_Can_FlexCan_ResetController(Controller) \ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-17056	Bug	[ADC] Put back the updates for the 3 tickets using a dummy branch<*>Detailed description (how to reproduce it): <*>Getting errors when merge code from a dev branch to develop. The problem is that now in develop, the update of there tickets (MCAL 16754, MCAL 16985 and MCAL 16991) is not contained. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>The update from these tickets is on develop. <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Put back the updates for the 3 tickets using a dummy branch
MCAL-17059	New	New Feature [EEP] Improve write in async mode Improve write in async mode to allow first Eep_MainFunction to write a page. Update state machine from Eep_Eeprom_WriteAsync to allow this.
MCAL-17084	New	New Feature [ADC] Searching through SW normal queue should be done from top of the queue, not from the end Searching through SW normal queue should be done from top of the queue, not from the end Currently, in Adc.c there is this kind of construnct in Adc_UpdateStatusReadGroupNoInt and Adc_UpdateStatusStopConversion : SwNormalQueueIndex = Adc_aUnitStatus[Unit].SwNormalQueueIndex - 1U; while (SwNormalQueueIndex < (Adc_QueueIndexType)ADC_QUEUE_INDEX_TYPE_MAXIMUM) { ..... SwNormalQueueIndex--;

Table continues on the next page...

ID	Subtype	Headline and Description
		} with this approach,the while loop will end after 0 value is checked and SwNormalQueueIndex rolls over to ADC_QUEUE_INDEX_TYPE_MAXIMUM value. This approach is not optimal, and even causes warnings in static analysis reports generated by some customers: "The loop is infinite or contains a run-time error. loop may fail due to a run-time error (maximum number of iterations: 20)" "Attempt to access to array element in range even values in [42 .. 208]. Valid index range: [0 .. 41]." "Conversion from int32 to unsigned int32 may overflow. Valid range: [0 .. 2 32 -1]"
MCAL-17098	Bug	[CAN] The Can_eDriverStatus do not declare when CanDevErrorDetection disabled <*>Detailed description (how to reproduce it): <*>When I run test with CanDevErrorDetection is disabled. I got an error at build as below: <*>identifier "Can_eDriverStatus" is undefined <*>Root cause: <*>In Can.c, we have condition for the code: <*>#if (CAN_DEV_ERROR_DETECT == STD_ON) <*>VAR(Can_StatusType, CAN_VAR) Can_eDriverStatus = CAN_UNINIT; <*>#endif <*>but the variable has called in Can_MainFunction_Mode function without the above condition. <*>FUNC(void, CAN_CODE) Can_MainFunction_Mode(void) <*>{ <*>/* Test whether the driver is already initialised. */ <*>if (CAN_UNINIT == Can_eDriverStatus) <*>{ <*>#if (CAN_DEV_ERROR_DETECT == STD_ON) <*>(void)Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_MODE, (uint8)CAN_E_UNINIT); <*>#endif /* (CAN_DEV_ERROR_DETECT == STD_ON) */ <*>} <*>Preconditions: <*>CanDevErrorDetection is disabled <*>Test Case ID (internal TC that caught the defect) - optional <*>Can_TS_02 <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-17099	Bug	[CAN] Missing exit exclusive area in Can_FlexCan_SetControllerToStopMode function<*>Detailed description (how to reproduce it): <*>Have some failed tests related to exclusive area () so I found an issue as below: <*>In Can_FlexCan_SetControllerToStopMode function, we have the section code as below: <*>SchM_Enter_Can_CAN_EXCLUSIVE_AREA_14(); <*>/*Clear flags for FIFO before delete content of FIFO*/ <*>REG_BIT_SET32(FLEXCAN_IFLAG1(u8HwOffset), (FLEXCAN_FIFO_OVERFLOW_INT_MASK_U32| FLEXCAN_FIFO_WARNING_INT_MASK_U32| FLEXCAN_FIFO_AVAILABLE_INT_MASK_U32)); <*>REG_BIT_SET32(FLEXCAN_IFLAG1(u8HwOffset), FLEXCAN_IFLAG1_BUF0I_U32); <*>SchM_Enter_Can_CAN_EXCLUSIVE_AREA_14(); <*>it's wrong because of missing the SchM_Exit_Can_CAN_EXCLUSIVE_AREA_14(); so it causes reentry_guard_14 failure. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>Can_TC_0005 <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-17101	Bug	[PORT] The driver should support all Unused GPIO Pin Termination<*>Imported from CQ ticket ENGR00390618. <*>Problem detailed description (how to reproduce it): <*>According to Application note AN5220, there ae some recommendation for "Unused GPIO Pin Termination" in page 30-"6.3 Unused GPIO Pin Termination" <*>http://www.nxp.com/assets/documents/data/en/application-notes/AN5220.pdf <*>For 176 or 100 pin device, the port driver does not support all implemented PAD yet. (0-148 for 176 pin device, 0-158 for 100 pin device). <*>According to this application note, the recommended configuration is (3) GPIO input with pull up/down or (4) GPIO output with pull up/down. <*>Therefore the port driver should support configure all implemented PADs (MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device). <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>(MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device) are not configured by port driver. <*>Expected behavior: <*>(MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device) are configured by port driver. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>AN5220 chapter -"6.3 Unused GPIO Pin Termination" <*>Proposed solution (Optional): <*>[...]
MCAL-17152	New	New Feature [GPT] Check and correct the guard for defines that should be enable/disable in Cfg.h Detailed description (how to reproduce it): Check and correct the unnecessary guard of defines that should be present in Cfg.h example: GPT_STM_ENABLECLOCKSWITCH file Gpt_Stm.c Observed behavior: the define is guarded in .c file to not generate compiler warnings and misra error Expected behavior: the correct way should be to be defined in Cfg.h and the warning is a check that Cfg file is incomplete. Proposed solution (Optional): check and correct for this for all platforms check also and specify in analysis the Requirement ID for those cases
MCAL-17158	New	New Feature [FLS] Add support for cache invalidations in QSPI Add support for cache operations(invalidations/clear) for QSPI. The internal memory mapped read buffers(AHB buffer and RX Buffer via AHB bus) are cacheable on PS cache. Add MCL Cache line api functions in the Fls_Qspi_SectorRead functions.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-17164	New	New Feature [PWM] Add support for running from User Mode NewWork Description: Add support for running from User Mode: 1) Create a PreCompile boolean configuration parameter in the plugin, named <Mdl>EnableUserModeSupport. The template for this parameter's description: "When this parameter is enabled, the MDL module will adapt to run from User Mode, with the following measures: (if applicable) a) configuring REG_PROT for ABC1, ABC2 IPs so that the registers under protection can be accessed from user mode by setting UAA bit in REG_PROT_GCR to 1 (if applicable) b) using 'call trusted function' stubs for all internal function calls that access registers requiring supervisor mode. (if applicable) c) other module specific measures for more information, please see chapter 5.7 User Mode Support in IM " Based on this confiugration parameter, a define will be created in the driver configuration with values STD_ON / STD_OFF, named <MDL>_ENABLE_USER_MODE_SUPPORT If this define is set to STD_ON, the driver code must: a) initialize REG_PROT for all IPs that have registers under REG_PROT protection b) implement the 'call trusted function' stubs for the functions that need supervisor mode access. (see the example below) (if applicable) c)implement other driver specific measures Otherwise, the driver will: a) make no configuration in REG_PROT () b) implement the 'call trusted function' stubs as simple defines to the associated functions 2) identify the list of internal driver functions that need supervisor mode access and enablethem to run from supervisor mode. There are some hardware registers that always need to be accessed from supervisor mode, and cannot be made accessible from user mode even with REG_PROT configuration. All functions that access these registers will need supervisor mode access. Example: You have identified that the function <Can_FlexCan_ResetController(uint8 Controller);> needs to be called from supervisor mode because it writes to CAN_MCR register. All calls to this function must be replaced with <Call_>Can_FlexCan_ResetController(Controller). The implementation of Call_Can_FlexCan_ResetController must be added in Can driver as: #if (STD_ON == CAN_ENABLE_USER_MODE_SUPPORT) #define Call_Can_FlexCan_ResetController(Controller) \\ Mcal_Trusted_Call(Can_FlexCan_ResetController,(Controller)); #else #define Call_Can_FlexCan_ResetController(Controller) \\ Can_FlexCan_ResetController(Controller); #endif This must be added in an internal header that is accessible in all driver files that need it.Also, Mcal.h from Base must be visible in this location. Mare sure that all identified functions like Can_FlexCan_ResetController are not defined as static or inline in the driver. The user application must be able to

Table continues on the next page...

ID	Subtype	Headline and Description
		call them from outside Mcal, if it contains an "extern" declaration. All these functions must be cleraly listed in the Analysis field of the ticket and in the driver IM, because they will need to be configured in the OS as trusted functions. Expected behavior: [?] Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-17190	Bug	[DIO] Fix the compiler warning for IMXVS4_42 <*>Detailed description (how to reproduce it): <*>There are some compiler warnings were raised when running test for CORE 1 of TreeRunner <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Remove the cast of pointer
MCAL-17191	Bug	[ICU] Update exclusive area for all IPVs<*>Now, exclusive areas in some IPvs overlap each other.Need update them correctly <*>Consider following rules: <*>Rule 4.1 All read-modify-write operations on global variables/structures fields shall be protected by an exclusive area, if another function/interrupt is updating that memory space. <*>Rule 4.2 All write operations on global variables/structures fields shall be protected by an exclusive area, if that operations may interfere with a read-modify-write operations on that global variables/structures fields. <*>Rule 4.3 For reentrant functions, global variables/ structures fields or local static variables read-modify-write operation shall be protected by an exclusive area. <*>Rule 4.4 All registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*), shall be protected by an exclusive area, if another function/interrupt is updating that register. <*>Rule 4.5 All write operations on registers shall be protected by an exclusive area, if that operation may interfere with a read-modify-write operation on that register. <*>Rule 4.6 For reentrant functions, registers read-modify-write operations (REG_BIT_CLEAR*, REG_BIT_SET*, REG_RMW*) shall be protected by an exclusive area. <*>Refer to attachment for more details
MCAL-17193	Bug	[SPI] Fix compiler warnings<*>Detailed description (how to reproduce it): <*>Fix compiler warnings in excel report file <*>Preconditions: <*>build with linaro compiler <*>Test Case ID (internal TC that caught the defect) - optional <*>Spi_TS_020_cfgPB1_CORE1 <*>Spi_TS_022_cfgCORE1 <*>Observed behavior: <*>No appear compiler warnings <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>- Comment for warning "cast to pointer from integer of different size [-Wint-to-pointer-cast]@9" <*>- initializer for field 'u32MCR' if SpiCsSelection tag does not exist for the warning "missing initializer for field 'u32MCR' of 'Spi_Ipw_DeviceAttributesConfigType' [-Wmissing-field-

Table continues on the next page...

ID	Subtype	Headline and Description
		initializers]@17"
MCAL-17198	Bug	[MCU] Misra rule 12.9 is raised when The values of McuMode and McuClockSettingId is 0<*>Detailed description (how to reproduce it): <*>[MCU] Misra rule 12.9 is raised when The values of McuMode and McuClockSettingId is 0 <*>Preconditions: <*>The values of McuMode and McuClockSettingId is 0 <*>Test Case ID (internal TC that caught the defect) - optional <*>Any <*>Observed behavior: <*>Misra rule 12.9 is raised <*>Expected behavior: <*>Misra rule 12.9 isn't raised <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>In ASR4.2.2, the range of McuMode and McuClockSettingId is 0~255. So, the values of Mcu_ModeConfigId and Mcu_ClkConfigId should be generated same with value in EB
MCAL-17419	New	New Feature [ICU] Analyze if ICU Signal Measurement works as intended in case of Overflow In Case of Overflow on FTM, see if the incorect measurement after overflow can be invalidated or corrected. In Signal Measurement Mode, the first duty / period computed after the overflow may be wrong.
MCAL-17431	Bug	[ADC] The static variable should not be declared inside functions<*>Problem detailed description (how to reproduce it): <*>The variable bFlag and CMRMask are declared as static variables inside Adc_ReadGroup and Adc_Adcdig_StartNormalConversion respectively <*>The customer tool reported following warning: <*>Missing MemMap-Pragma for variable "bFlag" in Adc.c, line 3428 and variable "CMRMask" in Adc_Adcdig.c, line 2621. <*>CE's comment: by default, the static variables will be stored in bss or data section. Depend on the linker file and pragma to re-name memory section of static variable, the compiler/linker warning may appear because of un-allocated section. If it is initialized at declaration, it is once time initialized. It is shared by multiple calls, if the function is re-entrance, its value may be mutual updated/ accessed from differenct tasks/interrupts then that could cause a malfuntion. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>The customer tool reported a warning concerning static variable inside a function. <*>Expected behavior: <*>No such above warning. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-17432	Bug	[PWM] Fix compier warning in Pwm_Ftm.c reported by treerunner RTM100<*>Detailed description (how to reproduce it): <*>The following compiler warnings need to be fixed: <*>warning #549-D: variable "eAlignment" is used before its value is set@82 <*>warning #549-D: variable "eIdleState" is used before its value is set@70 <*>Preconditions: <*>PWM_SET_OUTPUT_TO_IDLE_API is switched off and PWM_DE_INIT_API is switched on <*>Test Case ID (internal TC that caught

Table continues on the next page...

ID	Subtype	Headline and Description
		the defect) - optional <*>Pwm_TS_M05_cfg3_CORE1 <*>Pwm_TS_Eq_Cot_01_cfg5_CORE1 <*>Pwm_TS_Eq_Cot_01_cfg20_CORE1 <*>Pwm_TS_Eq_Cot_01_cfg1_CORE1 <*>Pwm_TS_Eq_Cot_01_cfg19_CORE1 <*>Pwm_TS_Eq_Cot_01_cfg16_CORE1 <*>Observed behavior: <*>Compile time generates warnings <*>The output state of the pins might be incorrect after Pwm_DeInit. <*>Expected behavior: <*>No compiler warning <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Guarding Pwm_FtmInternal_SetSwOutputControl(),eChannelIdleState by precompile defines have some gaps. <*>In order to have proper output state after Pwm_DeInit, and avoid compiler warnings, the workaround is to enable PWM_SET_OUTPUT_TO_IDLE_API.(PwmConfigurationOfOptApiServices/ PwmSetOutputToIdle) when Pwm_DeInit is used
MCAL-17444	New	New Feature [ETH] Create driver for S32K14X ASR 4.2 Please create driver for S32K14X ASR 4.2
MCAL-17470	New	New Feature [SPI] The array elements of struct type will build error with CodeWarrior compiler if not initialize size of array Detailed description (how to reproduce it): The array elements of struct type will build error with CodeWarrior compiler if not initialize size of array: - In struct Spi_AttributesConfigType on Spi_IPW_Types.h: + CONST(Spi_Ipw_ChannelAttributesConfigType, SPI_CONST) (* const pcChannelAttributesConfig)[]; + CONST(Spi_Ipw_DeviceAttributesConfigType, SPI_CONST) (* const pcDeviceAttributesConfig)[]; - In struct Spi_SequenceConfigType on Spi.h: + CONST(Spi_JobType, SPI_CONST) (* const pcJobIndexList)[]; - In struct Spi_JobConfigType on Spi.h: + CONST(Spi_ChannelType, SPI_CONST) (* const pcChannelIndexList)[]; - In struct Spi_ConfigType on Spi.h: + CONST(Spi_ChannelConfigType, SPI_CONST) (* const pcChannelConfig)[]; + CONST(Spi_JobConfigType, SPI_CONST) (* const pcJobConfig)[]; + CONST(Spi_SequenceConfigType, SPI_CONST) (* const pcSequenceConfig)[]; + CONST(Spi_HWUnitConfigType, SPI_CONST) (* const pcHWUnitConfig)[]; Preconditions: Build the code with CodeWarrior compiler. Test Case ID (internal TC that caught the defect) - optional All test case. Observed behavior: {code:java} ### mwccs12lisa.exe Compiler: # 695: &JOB_DSPI0_EXP_C0_1_ChannelAssignment_PC, /* List of Channels */ # Error: ^

Table continues on the next page...

ID	Subtype	Headline and Description
		# illegal implicit conversion from 'const unsigned char (*)[2]' to # 'const unsigned char (*const)[]' ### mwccs12lisa.exe Compiler: # 1489: &SEQ_DSPI02_2J_C0_2_JobAssignment_PC, /* List of Jobs */ # Error: ^ # illegal implicit conversion from 'const unsigned short (*)[2]' to # 'const unsigned short (*const)[]' ### mwccs12lisa.exe Compiler: # 1968: &S12spiChannelAttributesConfig_PC, # Error: ^ # illegal implicit conversion from 'const struct (*)[20]' to # 'const struct (*const)[]' ### mwccs12lisa.exe Compiler: # 2001: &SpiChannelConfig_PC, # Error: ^ # illegal implicit conversion from 'const struct (*)[20]' to # 'const struct (*const)[]' ### mwccs12lisa.exe Compiler: # 2002: &SpiJobConfig_PC, # Error: ^ # illegal implicit conversion from 'const struct (*)[29]' to # 'const struct (*const)[]' ### mwccs12lisa.exe Compiler: # 2003: &SpiSequenceConfig_PC, # Error: ^ # illegal implicit conversion from 'const struct (*)[27]' to # 'const struct (*const)[]' ### mwccs12lisa.exe Compiler: # 2005: &HWUnitConfig_PC, # Error: ^ # illegal implicit conversion from 'const struct (*)[1]' to # 'const struct (*const)[]' {code} Expected behavior: No error appear Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Re-define for the elements as below: - In struct Spi_AttributesConfigType on Spi_IPW_Types.h: + P2CONST(Spi_Ipw_ChannelAttributesConfigType, SPI_VAR, SPI_APPL_CONST) pcChannelAttributesConfig; + P2CONST(Spi_Ipw_DeviceAttributesConfigType, SPI_VAR, SPI_APPL_CONST) pcDeviceAttributesConfig; - In struct Spi_SequenceConfigType on Spi.h: + P2CONST(Spi_JobType, SPI_VAR, SPI_APPL_CONST) pcJobIndexList; - In struct Spi_JobConfigType on Spi.h: + P2CONST(Spi_ChannelType, SPI_VAR, SPI_APPL_CONST) pcChannelIndexList; - In struct Spi_ConfigType on Spi.h: + P2CONST(Spi_ChannelConfigType, SPI_VAR, SPI_APPL_CONST) pcChannelConfig; + P2CONST(Spi_JobConfigType, SPI_VAR, SPI_APPL_CONST) pcJobConfig; + P2CONST(Spi_SequenceConfigType, SPI_VAR, SPI_APPL_CONST) pcSequenceConfig; + P2CONST(Spi_HWUnitConfigType, SPI_VAR, SPI_APPL_CONST)

Table continues on the next page...

ID	Subtype	Headline and Description
		pcHWUnitConfig;
MCAL-17515	New	New Feature [MCU] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17516	New	New Feature [LIN] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144,

Table continues on the next page...

ID	Subtype	Headline and Description
		S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17517	New	New Feature [SPI] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17518	New	New Feature [I2C] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17519	New	New Feature [ADC] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17520	New	New Feature [DIO] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17521	New	New Feature [PORT] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A
MCAL-17522	New	New Feature [CRCU] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17523	New	New Feature [GPT] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144,

Table continues on the next page...

ID	Subtype	Headline and Description
		S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17524	New	New Feature [PWM] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17525	New	New Feature [ICU] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17526	New	New Feature [FLS] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17527	New	New Feature [FEE] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17528	New	New Feature [MCL] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A
MCAL-17529	New	New Feature [WDG] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17530	New	New Feature [EEP] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144,

Table continues on the next page...

ID	Subtype	Headline and Description
		S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17532	New	New Feature [CAN] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17533	New	New Feature [ETH] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17534	New	New Feature [RESOURCE] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17642	New	New Feature [Port] Missing PortPinMode in epd files Reported Baseline: BLN_SMCAL_4.0_MWCT101xS_RTM_1.0.1 Some PortPinModes such as LPSPI2, CAN2, LPUART2 are also not listed in

Table continues on the next page...

ID	Subtype	Headline and Description
		Port_mwct1014s_xxx.epd, Port_mwct1015s_xxx.epd and Port_mwct1016s_xxx.epd, making them unavailable to select from Tresos configuration.
MCAL-17661	New	New Feature [MCEM] Create driver IS FMEA for S32K14x RTM 1.0.0 ASR 4.2 NewWork Description: Create driver IS FMEA for S32K14x RTM 1.0.0 ASR 4.2. Requirement source: sMCAL Release criteria document version 5.1 http://compass.freescale.net/go/228798570 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17663	New	New Feature [MCEM] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - s32k146_lqfp64 (added) - s32k146_mapbga100 - s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17671	New	New Feature [MCEM] Create driver for S32K14X 4.2 (If Errata available, attach SMCAL Errata review template) NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Create detailed design for Platform Fix MISRA errors Create driver code (including xdm, cfg, ?) Fix VSMD errors Fix compiler warnings Add traceability markers in design and code (if it is a BETA release) Requirement source: - RM used ? version A - Errata used (fill Errata Review Template) ? version B (if needed) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17697	Bug	[CAN] Incorrect range for CanRxProcessing/CanTxProcessing/ CanWakeupProcessing/CanBusoffProcessing<*>Imported from CQ ticket ENGR00390907.<*>Problem detailed description (how to reproduce it): <*>Parameter CanRxProcessing/ CanTxProcessing/ CanWakeupProcessing/ CanBusoffProcessing has a value setting "Interrupt" or "Polling". But in Autosar, it is mentioned as INTERRUPT or POLLING. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>The range is different from ASR spec <*>Expected behavior: <*>The range follows ASR spec <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Update the range from "Interrupt" or "Polling" to "INTERRUPT" or "POLLING" and the template files accordingly.
MCAL-17705	Bug	[FLS] Incorrect implementation for FLS_TIMEOUT_HANDLING check<*>Imported from CQ ticket ENGR00390918.<*>Problem detailed description (how to reproduce it):<*>Reported baseline: BLN_SMCAL_4.0_PANTHER_RTM_2.0.0<*>The checking for FLS_TIMEOUT_HANDLING in Fls.c and Fls_Ac.c is not correctly implemented due to a missing of parenthesis.<*>Preconditions:<*>[...]<*>Test Case ID (internal TC that caught the defect) - optional<*>[...]<*>Trigger:<*>[...] <*>Observed behavior:<*>Missing parenthesis for FLS_TIMEOUT_HANDLING check <*>Expected behavior:<*>Update parenthesis for FLS_TIMEOUT_HANDLING check<*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)<*>Proposed solution (Optional): <*>Replace: #if FLS_TIMEOUT_HANDLING == STD_ON<*>by: #if (FLS_TIMEOUT_HANDLING == STD_ON)
MCAL-17706	New	New Feature [ETHTRCV] Add ethtrcv for S32k14x asr 4.2.2 NewWork Description: Add ethtrcv for S32k14x asr 4.2.2 Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): ethtrcv is added for s32k14x
MCAL-17711	New	New Feature [BASE] Add support for CSEC driver Compiler_cfg.h ->add comp abstraction keywords Base.mak (only for 4.2) ->add Csec_MemMap.h
MCAL-17752	New	New Feature [PWM] Implement errata e10856 for Mask 0N33V S32K142 and 0N57U S32K144 Original review ticket:-MCAL-17276- See e10856 in attachment and implemente the workaround. Note that S32K144 also impact by this errata Also please create ticket for base to enable the errata once this item is resolved
MCAL-17788	New	New Feature [MCU] Errors reported by configuration tolling should reference XDM parameters names Errors reported by configuration tolling should reference XDM parameters names precisely and not some other generic names used for those parameters. The purpose of the error information text is to describe which config parameters have invalid values. The text should contain the following information: - Exact parameter name(s) (as written in XDM) - Overlaying container (if container or parameter has a multiplicity different from 1) - invalid value and expected correct value. - (If possible) Reason why the value is invalid. - (If possible) Suggested (correct) value. See attachments for an such an error examples from CAN and PWM. In case of CAN, the issue is caused by the fact that the error text uses different names for the CAN Segments from the ones given in XDM. The solution is use the same names or labels as given by the XDM (i.e. PSEG1 -> CanControllerSeg1; PSEG2 -> CanControllerSeg2 and so on) In case of PWM, the issue is caused by the fact that error text pointing to channel number (i.e. channel "1") may have misleading interpretations (channel "1" may refer to either the index in the Channel List or to the specific 'ChannelId'). Also, for 4.0 platforms where different PwmChannelConfigSets may exist at the same time the value '1' may not be unique making identification of the channel even harder. A good solution would be to use the container "Name", since that value is unique across all variants, a better one would be to use both 'PwmChannel' container "Name" and 'PwmChannelConfigSet' name in which that channel was defined.
MCAL-17818	New	New Feature [BASE] Implement errata e10856 for Mask 0N33V S32K142 and 0N57U S32K144

Table continues on the next page...

ID	Subtype	Headline and Description
		Original review ticket:-MCAL-17276- See e10856 in attachment and implemente the workaround. Note that S32K144 also impact by this errata For Base driver: please create define for this errata
MCAL-17827	Bug	[CAN] Incorrect multiplicity of CanHwFilter container according to ASR 4.2 <*>Detailed description (how to reproduce it): <*>According to ASR 4.2.2 the container CanHwFilter should have multiplicity 0..*, but in the MCAL this has only multiplicity 1. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>CanHwFilter container has multiplicity 1 <*>Expected behavior: <*>CanHwFilter container should have multiplicity 0 .. *. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>This ticket needn't implement because when configure more CanHwFilter only help create more than configuration but when using them only a configuration. This is unnecessary.
MCAL-17829	Bug	[CAN] CAN_MEMORY_ECC_SUPPORT should be switched OFF<*>Detailed description (how to reproduce it): <*>Because ECC feature is not supported in CAN driver so CAN_MEMORY_ECC_SUPPORT should be defined as STD_OFF. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>CAN_MEMORY_ECC_SUPPORT is set to STD_ON because of Can.CanConfig.MemoryECC:STD_ON defined in resource. <*>Expected behavior: <*>CAN_MEMORY_ECC_SUPPORT should be set to STD_OFF <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-17878	Bug	[CAN] Mismatch between #include call in header files and the real filename of those includes<*>Detailed description (how to reproduce it): <*>Some of the includes in the MCAL modules do not match the filenames they include. Apparently, this was tolerated in a Windows build environment, but under Linux, this bug makes itself visible. For example: <*>In Can_Flexcan.h, Reg_eSys_Flexcan.h is included. But the actual filename is Reg_eSys_FlexCan.h (_different in letter "C" in _Flexcan.h vs _FlexCan.h_) <*>Attachment contains differences found. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>Different between #include of header file and the real name of the header file. <*>Expected behavior: <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-17879	Bug	[SPI] Mismatch between #include call in header files and the real filename of those includes<*>Detailed description (how to reproduce it): <*>Some of the includes in the MCAL modules do not match the filenames they include. Apparently, this was tolerated in a Windows build environment, but under Linux, this bug makes itself visible. For example: <*>In Can_Flexcan.h, Reg_eSys_Flexcan.h is included. But the actual filename is

Table continues on the next page...

ID	Subtype	Headline and Description
		Reg_eSys_FlexCan.h (_different in letter "C" in _Flexcan.h vs _FlexCan.h_) <*>Attachment contains differences found. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Different between #include of header file and the real name of the header file. <*>Expected behavior: <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-17880	Bug	[WDG] Mismatch between #include call in header files and the real filename of those includes<*>Detailed description (how to reproduce it): <*>Some of the includes in the MCAL modules do not match the filenames they include. Apparently, this was tolerated in a Windows build environment, but under Linux, this bug makes itself visible. For example: <*>In Can_Flexcan.h, Reg_eSys_Flexcan.h is included. But the actual filename is Reg_eSys_FlexCan.h (_different in letter "C" in _Flexcan.h vs _FlexCan.h_) <*>Attachment contains differences found. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>Different between #include of header file and the real name of the header file. <*>Expected behavior: <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-17889	Bug	[FLS] Missing MemMap sections in Fls_Flash.c file<*>Detailed description (how to reproduce it): <*>MemMap sections are missing from Fls_Flash.c file. <*>Expected behavior: <*>Function declarations and definitions should be guarded by specific MemMap sections. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Add #define FLS_START_SEC_CODE and #define FLS_STOP_SEC_CODE for both function declaration and definitions Fls_Flash.c file.
MCAL-17891	New	New Feature [MCU] Update change according to new Reference manual (Rev.4) for S32K14X RTM 1.0.0 ASR 4.2 NewWork Description: In the last S32K14X 4.2 beta 0.9.0 release, the reference manual for cut1.0 was reviewed in this ticket: MCAL-10932. After that, the Reference manual Rev. 3 was reviewed by almost drivers in this ticket: MCAL-15836. Currently, the reference manual Rev.4 is available. Please review this new reference manual provided in attachment compared to your latest version you reviewed in the past. The differences between 2 versions should be recorded clearly in Analysis tab. Please double check in your 4.2 code base to see if anything needs to be done on your code according to this new reference manual but has not been done yet, then create new follow-up ticket to implement this change. Requirement source: S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf,

Table continues on the next page...

ID	Subtype	Headline and Description
		Errata.pdf...) Proposed solution (Optional): - The New Feature ticket used for HW Reference Manual will be closed with Resolved and the result of the HW Reference Manual review shall be recorded in the analysis report, proposed solution section. - If a software change need to be implemented, a new ticket (Bug or New Feature) is raised. This new ticket shall contain the New Work CR used for HW Reference Manual review in description for reference.
MCAL-17918	New	New Feature [I2C] Fix MISRA violations for S32K14X There are some MISRA violations in the configuration files which must be fixed.
MCAL-17919	New	New Feature [I2C] Fix VSMD warnings for S32K14X ASR 4.2 There is a VSMD warning regarding "I2CClockRef" which states that it has "dynamic" multiplicity but no "ECUC-MULTIPLICITY-CONFIGURATION-CLASS" elements.
MCAL-17930	New	New Feature [EEP] Follow up fix code review against checklist for S32K14x See more details in attachment tab
MCAL-17931	Bug	[GPT] Incorrect syntax [!ERROR] in VersionCheck_Src.m files<*>Detailed description (how to reproduce it): <*>Problem is encountered in ".\generate_PC \Gpt_VersionCheck_Src.m" and ".\generate_PB\Gpt_VersionCheck_Src_PB.m" in which, [!ERROR] syntax for difference checking between ArReleaseRevisionVersion of the Basic Software Module Description and the Code template file are incorrectly implemented. <*>[!IF "not(num:i(ArReleaseRevisionVersion) = num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE))"!] <*>*{color:#d04437}[!ERROR]{color}* <*>"AUTOSAR release revision version number of the Basic Software Module Description file (Gpt.epd version [!"ArReleaseRevisionVersion"!]) and the Code template file (Gpt_Cfg.c version [!"num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE)"!]) are different" <*>[!ENDERROR!] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>Missing symbol "!" at the end of [!ERROR] <*>Expected behavior: <*>Correct the syntax <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Replace: <*>[!IF "not(num:i(ArReleaseRevisionVersion) = num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE))"!] <*>{color:#ff0000}*[!ERROR]*{color} <*>"AUTOSAR release revision version number of the Basic Software Module Description file (Gpt.epd version [!"ArReleaseRevisionVersion"!]) and the Code template file (Gpt_Cfg.c version [!"num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE)"!]) are different" <*>[!ENDERROR!] <*>by: <*>[!IF "not(num:i(ArReleaseRevisionVersion) =

Table continues on the next page...

ID	Subtype	Headline and Description
		num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE))"!] <*>{color:#14892c}*[!ERROR!]*{color} <*>"AUTOSAR release revision version number of the Basic Software Module Description file (Gpt.epd version [!"ArReleaseRevisionVersion"!]) and the Code template file (Gpt_Cfg.c version [!"num:i($GPT_AR_RELEASE_REVISION_VERSION_TEMPLATE)"!]) are different" <*>[!ENDERROR!]
MCAL-17951	New	New Feature [OCU] Create driver for S32K14X 4.2 (If Errata available, attach SMCAL Errata review template) NewWork Description: Create detailed design for Platform Fix MISRA errors Create driver code (including xdm, cfg, ?) Fix VSMD errors Fix compiler warnings Add traceability markers in design and code (if it is a BETA release) Requirement source: - RM used ? version A - Errata used (fill Errata Review Template) ? version B (if needed) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17954	New	New Feature [OCU] Create driver IS FMEA for S32K14x RTM 1.0.0 ASR 4.2 NewWork Description: Create driver IS FMEA for S32K14x RTM 1.0.0 ASR 4.2. Requirement source: sMCAL Release criteria document version 5.1 http://compass.freescale.net/go/228798570 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-17963	New	New Feature [PWM] implement middle point reload for FTM PR--MCAL-3309-.pwm now is applied for FTM: The Pwm driver shall allow selection of the reload point for center aligned PWM in complementary mode (see PR--MCAL-3213-); possible choices shall be `middle of period? and `end of period?.
MCAL-17975	New	New Feature [GPT] User Manual for each driver should contain a "How To" configure chapter for advanced features All drivers contain a set of features which are not (fully) described by

Table continues on the next page...

ID	Subtype	Headline and Description
		AUTOSAR SWS (i.e NON-ASR features). The UM's for each driver should contain a dedicated chapter for show-casing how certain features need to be configured/enabled. Add the information into 3.6 Driver usage and configuration tips For example: - HW Triggers and interface between PWM, MCL and ADC (each driver should describe *it's own needs* to enable this feature); - DMA (memory to memory should be exemplified in MCL, however special uses-cases like ICU or ADC should contain theirs own examples); - Fault handling, synchronous wave-form updating and use of center aligned wave-form in PWM; - Flash sectors (with start addresses and sizes) and tip-and-tricks on how to enable them to avoid flash access exceptions. - MCU mode managements examples. The use-case here is to show users how to enable/disable certain clocks in different run-modes to reduce power consumption. Note. Last entry in this list can only describe some examples and not an absolute use-case since mode-managements and power consumption are dictated by user-system requirements and are specific for each user-project. The purpose of those examples is to show one use-case which definitely works.
MCAL-17976	New	New Feature [GPT] [UM] The chapter deviation from requirements shall be updated NewWork Description: Check that the requirements available in the chapter Deviation from requirements include the requirements that: - do not have the platform name in the list of platforms (N/S) - have the platform name in the list of platforms and are not fulfilled in (N/I) Requirement source: [?] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
MCAL-17998	New	New Feature [ADC] Update Dev tests for S32K14x Update dev tests to support all derivatives for s32k14x 4.2
MCAL-18003	New	New Feature [ICU] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ?Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return:

Table continues on the next page...

ID	Subtype	Headline and Description
		False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18004	New	New Feature [WDG] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ?Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status

Table continues on the next page...

ID	Subtype	Headline and Description
		False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18006	New	New Feature [MCEM] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ? Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function)

Table continues on the next page...

ID	Subtype	Headline and Description
		Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18007	New	New Feature [OCU] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ?Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18011	New	New Feature [SPI] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ? Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18016	New	New Feature [FEE] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ? Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms.

Table continues on the next page...

ID	Subtype	Headline and Description
		1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18017	New	New Feature [CAN] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ?Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter.

Table continues on the next page...

ID	Subtype	Headline and Description
		It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry. This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18019	New	New Feature [EEP] Support for adding all types of event IDs In order to separate the implementation/interactions between any driver and the DEM functionality(on ASR 4.2), the following functions were implemented in ? Dem_stub.c?(file placed in build_env). Any test should use the following functions in order to test the DEM events(the test should NOT access directly the internal variables from DEM). The tests should be updated in order to support the new ASR4.2 approach. These updates affects only ASR4.2 platforms. 1. Dem_TestNoError(void) ? (already exists in this file) It will parse all events stored so far. It will return: False ? if any PREFAILED or FAILED event is found True ? if no PREFAILED or FAILED event is found This function will also reset all the events, after the above check (NumberOfEvents=0) 2. Dem_TestLastReportErrorStatus(EventId, EventStatus) - (already exists in this file) It will search the buffer of events, from the last event logged to the first one(from the newest to the oldest event), for the event ID&Status passed as parameter. It will return: True ? if the event is found (the first one found) with the requested status False ? if that event is not found in the entire buffer with the requested status This function will also delete the event found ((NumberOfEvents--) && (all events will be shifted with one position in the buffer)) 3. Dem_ClearEvents(void) ? (new function) It will reset all the events (NumberOfEvents=0) 4. Dem_GetEvent(IndexNumber, &EventId, & EventStatus) ? (new function) It will return the event found on the index passed as parameter(IndexNumber). This function will use the parameters 2 and 3(&EventId, & EventStatus) to return that entry.

Table continues on the next page...

ID	Subtype	Headline and Description
		This function will return 1 if the index is OutOfRange. 5. Dem_GetEventCount(void) ? (new function) Returns the NumberOfEvents that are logged so far. 6. Dem_BufferOverflow(void) ? (new function) Returns: True ? if the number of events logged so far exceeds the DEM event buffer size elements False ? if there is still room in the buffer In case of Overflow: the last position (Dem_EventId[255])will be overwritten if the new event is a PREFAILED or FAILED. This way, you will always have the last error reported. And together with the function Dem_BufferOverflow, you will know if you have the entire/correct sequence of events reported to DEM.
MCAL-18023	New	New Feature [PORT] Implement the changes after review new Reference manual (Rev.4) for S32K14X RTM 1.0.0 ASR 4.2 NewWork Description: In the last S32K14X 4.2 beta 0.9.0 release, the reference manual for cut1.0 was reviewed in this ticket: MCAL-10932. After that, the Reference manual Rev. 3 was reviewed by almost drivers in this ticket: MCAL-15836. Currently, the reference manual Rev.4 is available. Please review this new reference manual provided in attachment compared to your latest version you reviewed in the past. The differences between 2 versions should be recorded clearly in Analysis tab. Please double check in your 4.2 code base to see if anything needs to be done on your code according to this new reference manual but has not been done yet, then create new follow-up ticket to implement this change. Requirement source: S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - The New Feature ticket used for HW Reference Manual will be closed with Resolved and the result of the HW Reference Manual review shall be recorded in the analysis report, proposed solution section. - If a software change need to be implemented, a new ticket (Bug or New Feature) is raised. This new ticket shall contain the New Work CR used for HW Reference Manual review in description for reference.
MCAL-18024	Bug	[SPI] Redundant token NOCODE in Spi_RegOperation.m<*>Detailed description (how to reproduce it): <*>Unexpected token [!ENDMACRO!]? line 1122 in Spi_RegOperation.m (see the attached picture) <*>Preconditions: <*>Use the label: BLN_SPI_022 <*>Test Case ID (internal TC that caught the defect) - optional <*>Spi_TS_001 <*>Observed behavior: <*>Unexpected token [!ENDMACRO!]? line 1122 in Spi_RegOperation.m on EB Tresos <*>Expected behavior: <*>No error appears <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>In Spi_RegOperation.m: Remove redundant token [!NOCODE!] in the Macro [!MACRO "Spi_ConfigurationInfo"!]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18025	New	New Feature [I2C] Fix tabulators and duplicated UUIDs on S32K14X Tabulators: ||File||Hits|| |I2C_Cfg_42.h|1| Duplicated UUIDs tags: ||UUID||Location|| |ECUC:087f3523-f919-4b2d-841a-d18846908390|I2C.xdm:1616| |ECUC:f9793957-805b-400b-921f-acc6d3289e14|I2C.xdm:94| |ECUC:fdac1ad9-1217-44a8-bc3f-341635a20162|I2C.xdm:1677| |ECUC:5419eaab-b72b-4ee1-9e89-22e203491b61|I2C.xdm:802|
MCAL-18026	Bug	[FLS] Add Det.h and Dem.h include in Fls_IPW.c for S32K<*>Detailed description (how to reproduce it): <*>Fls_IPW.c specific file for S32K does not have the #include "Det.h", "Dem.h" for the DET/DEM errors in Fls_IPW_AbortSuspended() function. <*>Preconditions: <*>DET/DEM error reporting enabled <*>Test Case ID (internal TC that caught the defect) - optional <*>SampleApp <*>Observed behavior: <*>Build error <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>* Fix the ifelse(M4...) parameter check error from Fls_Types.h <*>* Fix CRC error. <*>- Remove the "ifelse(M4_SRC_USED_PERIPHERAL,`IPV_FTFE',`dnl" construct, this IP is always present in this specific file. <*>- Add the following into Fls_IPW.c for S32K: <*>#if (STD_ON == FLS_INTERNAL_FLASH_AVAILABLE) <*>#if (FLS_INTERNAL_SECTORS_CONFIGURED == STD_ON) <*>ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_0_REV_0003',`dnl <*>#if(FLS_DISABLE_DEM_REPORT_ERROR_STATUS == STD_OFF) <*>#include "Dem.h" <*>#endif <*>',`dnl <*>#if(FLS_RUNTIME_ERROR_DETECT == STD_ON) <*>#include "Det.h" <*>#endif <*>')dnl <*>#endif <*>#endif
MCAL-18028	New	New Feature [RESOURCE] Add support for Mcem and Ocu MCEM and Ocu have some resource file. It is required to update Resource.cfg to compile also resource for Mcem and Ocu modules.
MCAL-18029	New	New Feature [ADC] AdcSampleTimeDuration can support sample time of 2 to 256 ADC clock cycles Detailed description (how to reproduce it): Currently driver only support sample time of 2 to 255 ADC clock cycles, however it can support sample time of 2 to 256 as RM said. Proposed solution (Optional): [...] ? Detailed description (how to reproduce it): [...] Preconditions: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: Support sample time of 2 to 256 ADC clock Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18031	Bug	[SPI] Issue with Spi_RegOperations.m file<*>Detailed description (how to reproduce it): <*>Error at build when porting test case use job end notifications is enable. <*>Build fail with case use job end notifications is enable. Template file do not generate expect output <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Spi_RegOperation.m: Macro "Spi_ListOfNotificationFunctions" is missing token [!CODE!] and [!ENDCODE!], so Job Notification functions will not be generated. <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Add token [!CODE!] and [!ENDCODE!] into macro "Spi_ListOfNotificationFunctions"
MCAL-18033	Bug	[ICU] Fix wrong define user interrupt for FTM<*>Detailed description (how to reproduce it): <*>FTM user interrupt define need combine be pair channel because they same interrupt vector. <*>example: ICU_FTM_1_CH_0_CH_1_ISR_USED, ICU_FTM_1_CH_2_CH_3_ISR_USED <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>ICU_FTM_1_CH_0_ISR_USED <*>ICU_FTM_1_CH_1_ISR_USED <*>.... <*>Expected behavior: <*>ICU_FTM_1_CH_0_CH_1_ISR_USED <*>ICU_FTM_1_CH_2_CH_3_ISR_USED <*>... <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>add case to process it in marcro
MCAL-18034	New	New Feature [OCU] Update resource according to new Reference manual Rev.4 for S32K14X ASR 4.2 NewWork Description: Please update resource according to latest Reference manual to support the following derivatives. Refer to MCAL-17493 for more detail about Reference manual. - s32k148_lqfp144 - s32k148_lqfp176 - s32k148_mapbga100 - s32k146_lqfp144 - s32k146_lqfp100 - *s32k146_lqfp64 (added)* - s32k146_mapbga100

Table continues on the next page...

ID	Subtype	Headline and Description
		- s32k144_lqfp100 - s32k144_lqfp64 - s32k144_mapbga100 - s32k142_lqfp100 - s32k142_lqfp64 Note: This release will support to test successfully for S32K142, S32K144, S32K148 derivative. For S32K146 derivative, the resource should be available and ready for use only. It is no need to test this time for S32K146. Requirement source: - S32K1xx reference manual: Rev.4, 06/2017: http://compass.freescale.net/go/235438173 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A
MCAL-18035	Bug	[CAN]Update code driver for pretended networking<*>Detailed description (how to reproduce it): <*>When Pretended Networking is enabled, build will be failed because it contains some define which haven't declared yet. <*>Preconditions: <*>- Pretended Networking is enabled. <*>Test Case ID (internal TC that caught the defect) - optional <*>TS 90, 91, 92 <*>Observed behavior: <*>Replace these define by a define equivalent. <*>Expected behavior: <*>Build pass. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Replace these define by a define equivalent.
MCAL-18037	New	New Feature [ICU][S32K14X_4.2] Add support Requirement PR-MCAL-3242.icu Detailed description (how to reproduce it): {color:#ff0000}PR--MCAL-3242-.icu{color} An additional supplier specific, Boolean configuration parameter shall be introduced for provision of additional/alternative behavior. The default value of this configuration parameter shall be ?FALSE?. This additional but optional functionality shall comprise: Additional API "Icu_LevelType Icu_GetInputLevel (uint8 Channel)" to read the level of an ICU channel input pin. The return value shall be 1(ICU_LEVEL_HIGH) or 0(ICU_LEVEL_LOW). Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove read-only IcuGetInputLevelApi in file config

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18043	New	New Feature [CAN] Update development test case for S32K14X 4.2 Detailed description (how to reproduce it): Update development test case Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18045	Bug	[GPT] GptWakeupSourceRef was not editable<*>Detailed description (how to reproduce it): <*>- Cannot edit GptWakeupSourceRef when GptWakeupConfiguration was enabled. <*>Preconditions: <*>- GptEnableWakeup and GptWakeupConfiguration were enabled. <*>Test Case ID (internal TC that caught the defect) - optional <*>- tc_fnc_gpt_00105 <*>Observed behavior: N/A <*>Expected behavior: N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): Repair GptWakeupSourceRef node in Gpt.xdm
MCAL-18047	Bug	[ADC] Update Adc_Adc12bsarv2_StopConversionCheckTimeout function<*>Detailed description (how to reproduce it): <*>In Adc_Adc12bsarv2_StopConversionCheckTimeout function, this line should be updated: <*>REG_WRITE32(REG_READ32(ADC12BSARV2_SC1_REG_ADDR32(Unit, (uint32)u8SCRegister)), ADC12BSARV2_ADCH_CONVERSION_DISABLE_U32); <*>become: <*>REG_WRITE32(ADC12BSARV2_SC1_REG_ADDR32(Unit, (uint32)u8SCRegister), ADC12BSARV2_ADCH_CONVERSION_DISABLE_U32); <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A
MCAL-18048	Bug	[PWM] Add fixed shifted period to PwmChannelClass<*>Detailed description (how to reproduce it): <*>PwmChannelClass lacked PWM_FIXED_PERIOD_SHIFTED <*>Preconditions: <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>PwmChannelClass lacked PWM_FIXED_PERIOD_SHIFTED <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention

Table continues on the next page...

ID	Subtype	Headline and Description
		also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Add this class
MCAL-18049	Bug	[ADC] Should not disable hardware in the loop<*>Detailed description (how to reproduce it): <*>Currently in _StopConversionCheckTimeout function, adc hardware is disabled under the loop. <*>Hardware can be stuck. <*>So we should disable one time and wait until hardware change state <*>Preconditions: <*>N/A <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A
MCAL-18050	Bug	[PORT] Inconsistent/Incorrect usage in memory allocation sections<*>Detailed description (how to reproduce it): <*>Port build fail with IAR <*>"D:\Git_view\Git_View\SASW\JSW_MCAL\XPC56xx\output\S32K14X_S32K146\icu\tse_bbx_wir_icu_00100\generate\src\Port_VS_2_PBcfg.c",332 Error[Pe147]: declaration is incompatible with "Port_ConfigType const Port_Runtime_VS_2 @ ".mcal_const"" (declared at line 883 of "../../../SASW/JSW_MCAL/XPC56xx/output/S32K14X_S32K146/icu/tse_bbx_wir_icu_00100/generate/include/Port_Cfg.h")
MCAL-18053	New	New Feature [PWM] switch off ReloadNotification for unsupported platforms Caracssonne+ was developed with new feature: Reload Notification. This feature was guarded with a precompile define check. This define should to switch OFF for unsupported platforms
MCAL-18056	New	New Feature [MCEM] Allow use check bit to inject fault Detailed description (how to reproduce it): According the Reference Manual for S32K14x: Error Injection Module (EIM) supports error injection to Check Bit field but MCEM driver did not support. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18057	New	New Feature [MCEM] Allow inject fault with all nit positions Detailed description (how to reproduce it): According RM of S32K144x, EIM support 32 bits in WORD1 register in order to support inject fault to 32 bit position, but driver still support to inject fixed bit position (bit 0 and bit 1) in WORD1 register. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: [...] Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18058	Bug	[MCEM] Driver did not implement two-stage enablement mechanism<*>Detailed description (how to reproduce it): <*>The EIM provides protection against accidental enabling and reconfiguration of the error <*>injection function by enforcing a two-stage enable mechanism. To properly enable <*>the error injection mechanism for a channel: <*>? Write 1 to the EICHEN[EICHnEN] field, where n denotes the channel number. <*>? Write 1 to EIMCR[GEIEN]. <*>But in current implementation of MCEM just using "ONE"-stage enable mechanism by using only: REG_WRITE32(EIRM_EIM_EICHEN_ADDR32, EIRM_EIM_EICHEN_CH0_U32); <*>see: Mcem_Eirm_InjectFault function. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18061	Bug	[OCU] Adding some resource for S32K14X 4.2<*>Detailed description (how to reproduce it): <*>Missing some resource FTM4 and FTM7 on all both DERIVATIVE S32K146 and S32K148 <*>FTM4 : Missing FTM_4_CH_1,FTM_4_CH_2,FTM_4_CH_3,FTM_4_CH_4 ,FTM_4_CH_7.. (S32K146 & S32K148) <*>FTM7: Missing FTM_7_CH_4 ,FTM_7_CH_6.. (S32K148) <*>
MCAL-18062	Bug	[ADC] Incorrect channel's name of VS_>1 <*>Detailed description (how to reproduce it): <*>When you config channel for each VS0,VS2. Group8_VS0 : /Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_0 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_1 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_0 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_0 <*>for <*>Group8_VS2 : /Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_0 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/

Table continues on the next page...

ID	Subtype	Headline and Description
		AdcChannel_1 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_2 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_0 <*>/Adc/Adc/AdcConfigSet/AdcHwUnit_0/AdcChannel_2 <*>The channel's name of group always was definded as channel's name of VS_0. <*>
MCAL-18063	Bug	[PWM]Remove manual fault clearing option for FTM<*>Detailed description (how to reproduce it): <*>Manual Fault clearing is selectable but has not implemented yet. To implement this feature, it should provide 2 APIs to check fault flag and clear flag, this is called polling mode. <*>Currently, just remove Manual Fault clearing <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-18064	New	New Feature [PWM] Extend deadtime range for S32K Kinetis has FTM_DEADTIME[DTVALEX] which allows extend deatime counter 4 bits more Treerunner does not.
MCAL-18065	Bug	[MCL][FTM] Fail at build in Ftm_Common.c file<*>Detailed description (how to reproduce it): <*>## <*>"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 694: error #7: <*>unrecognized token <*>* @brief Independent interrupt handler. <*>^ <*>"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 694: error #40: <*>expected an identifier <*>* @brief Independent interrupt handler. <*>^ <*>"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 695: error #7: <*>unrecognized token <*>* @details Interrupt handler for FTM module 0 channel 4 - channel 5 <*>^ <*>"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 697: error #7: <*>unrecognized token <*>* @isr <*>^ <*>### <*>It make some module whichs use FTM ipv fail at build <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...] ? Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18068	New	New Feature [MCL] Extend deadtime range for S32K Kinetis has FTM_DEADTIME[DTVALEX] which allows extend dead-time of counter to 4 bits more Treerunner doesn't have this feature.
MCAL-18071	Bug	[OCU] S32K FTM version should support PWMEN bits<*>Detailed description (how to reproduce it): <*>Ocu driver does not have output. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Output should work as designed on OCU. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>For. IPV_FTM_05_00_04_00 enable PWMEN bits after Ocu_Ftm_Init
MCAL-18072	Bug	[OCU] Ocu_Ftm_ProcessCommonInterrupt declared implicitly <*>Detailed description (how to reproduce it): <*>when not configuration OcuNotificationSupported and then driver compile failed... <*>log file : <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_0); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 633: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_1); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 667: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_2); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 686: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_3); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 720: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_4); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 739: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared

Table continues on the next page...

ID	Subtype	Headline and Description
		implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_5); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 773: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_6); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 792: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_0_CH_7); {color} <*>{color:#000000}^ {color} <*>{color:#FF0000}"c:/vv_tools/eb/EB_tresos_Studio_23.0.0_b170330-0431_00/plugins/Mcl_TS_T40D2M10I0R0/src/Ftm_Common.c", line 874: error #223-D: {color} <*>{color:#000000}function Ocu_Ftm_ProcessCommonInterrupt declared implicitly {color} <*>{color:#000000}Ocu_Ftm_ProcessCommonInterrupt((uint8)FTM_1_CH_0); {color} <*>..... <*>{color:#000000}.....{color}
MCAL-18074	Bug	[ICU] [S32K14x 4.2] Improvement resource <*>Detailed description (how to reproduce it): <*>Wrong number module FTM in Resource <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Correct number module FTM in Resource <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18075	New	New Feature [ADC] Update driver follow the change of prototype in Mcl module I?ve made change for Mcl driver, and it will impact to other driver and testing of course. In previous version, MCL APIs for the TCD address is sometimes used as uint32, other times as pointer to uint32, the APIs should be updated to have a consistent approach. so for now: For TCD address it will only use Mcl_DmaTcdType, so you don?t need to cast like this : Mcl_DmaTcdSetIterCount((Mcl_DmaTcdType*)tcd_address,ITER_COUNT_ELINK_NO); And now it?s like: Mcl_DmaTcdSetIterCount((Mcl_DmaTcdType)tcd_address,ITER_COUNT_ELINK_NO); Please try with PVT_MCL_SMCAL_4.2_S32K14X_CHANGE_TCD_ADD_TYPE_V1 and let me know if you find any problem.
MCAL-18078	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[MCU] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18081	New	New Feature [I2C] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented.

Table continues on the next page...

ID	Subtype	Headline and Description
		It should be documented in the missing ones. The information should be present in IM only. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18082	New	New Feature [ADC] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos.

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18083	New	New Feature [DIO] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard:

Table continues on the next page...

ID	Subtype	Headline and Description
		- <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18084	New	New Feature [PORT] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant.

Table continues on the next page...

ID	Subtype	Headline and Description
		Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18085	New	New Feature [CRCU] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18086	New	New Feature

Table continues on the next page...

ID	Subtype	Headline and Description
		[PWM] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18087	New	New Feature [GPT] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be

Table continues on the next page...

ID	Subtype	Headline and Description
		present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18088	New	New Feature [ICU] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18089	New	New Feature [OCU] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all

Table continues on the next page...

ID	Subtype	Headline and Description
		parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18095	New	New Feature [CAN] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18096	New	New Feature [ETH] Update the UM/IM to reflect the deviation from standard for Pre-Compile Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18097	New	New Feature [MCEM] Update the UM/IM to reflect the deviation from standard for Pre-Compile

Table continues on the next page...

ID	Subtype	Headline and Description
		Imported from CQ ticket ENGR00392106. Problem detailed description (how to reproduce it): Problem description from Customer Engineer: - In some driver like ADC, SPI, etc, the deviation from standard for variant aware parameters and not variant aware parameters is documented in the Integration Manual - In some driver like MCU, PORT, the deviation is documented in User Manual. - In some other drivers like Fr, Gpt, Pwm, Icu, the deviation is not documented. *It should be documented in the missing ones. The information should be present in IM only.* Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The document does not provide sufficient infos. Expected behavior: In the integration manual for each driver in the chapter 3.2 files required for compilation the following information should be present for all asr 4.2 platforms: <Driver> Generated Files (list all files generated by the driver) For driver compilation, <Driver>_<VariantName>_PBcfg.c should be generated by the user using a configuration tool. The file contains the definition of the init pointer for the respective variant. As a deviation from standard: - <Driver>_<VariantName>_PBcfg.c files will contain the definition for all parameters that are variant aware, independent of the configuration class that will be selected (PC, LT, PB) - <Driver>_Cfg.c file will contain the definition for all configuration structures containing only variables that are not variant aware, configured and generated only once. This file alone does not contain the whole structure needed by <Mdl>_Init function to configure the driver. Based on the number of variants configured in the EcuC, there can be more than one configuration structure for one module even for PreCompile variant. Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
MCAL-18099	Bug	[MCEM] Mcem_Init() must clear all error flags<*>Detailed description (how to reproduce it): <*>In order to driver works more perfect and avoid some unexpected problem, Mcem_Init() must reset injection register, flag before doing configuration. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18104	Bug	[LIN] Break length was wrong initial for PB configuration for S32K14X ASR

Table continues on the next page...

ID	Subtype	Headline and Description
		4.2<*>Detailed description (how to reproduce it): <*>In CONST(Lin_ChannelConfigType,LIN_CONST) Lin_xxx config (in Lin_PBcfg.c file), value of break length was wrong calculate: <*>[!VAR "IntegerBrkLength" = "substring-after(BreakLength,'_')"!] <*>[!IF "$IntegerBrkLength = 36"!] <*>[!VAR "VarBreakLength"="14"!] <*>[!ELSEIF "$IntegerBrkLength = 50"!] <*>[!VAR "VarBreakLength"="15"!] <*>[!ELSE!] <*>[!VAR "VarBreakLength"="$IntegerBrkLength - 10"!] <*>[!ENDIF!] <*>[!ENDNOCODE!][!// <*>[!"num:inttohex($VarBreakLength,2)"!]U /* BreakLength = [!"BreakLength"!] bits */ <*>Example: When break length =13 bits, return value is 3U, instead is 13U. <*>Preconditions: <*>Call Lin_Init() function <*>Test Case ID (internal TC that caught the defect) - optional <*>tc_wir_lin_00100.c <*>Observed behavior: <*>Initial wrong break length <*>Expected behavior: <*>Initial true break length <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Replace the above block by: <*>[!VAR "IntegerBrkLength" = "substring-after(BreakLength,'_')"!] <*>[!ENDNOCODE!][!// <*>[!"num:inttohex($IntegerBrkLength,2)"!]U /* BreakLength = [!"BreakLength"!] bits */
MCAL-18107	Bug	[MCEM] Mcem interrupt routine must disable when disabling interrupt source<*>Detailed description (how to reproduce it): <*>In current implementation for ISR: MCEM_EIRM_FAULT_SINGLEBIT_ISR or MCEM_EIRM_FAULT_DOUBLEBIT_ISR are still defined even corresponding interrupt channel is OFF. <*>Note: when all interrupt is OFF, it is no meaning to use Mcem Error Notification with Error Address API, so in this case the filed Mcem Error Notification must disable. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18108	Bug	[OCU] Ocu_Ftm_DeInit should only clear the channels used in a configuration<*>Detailed description (how to reproduce it): <*>When more channels are configured in OcuHWSpecificSettings than there are in OcuChannel. some wrong addressed will be written in Ocu_Ftm_DeInit. <*>The loops for DeInit is limited to OCU_FTM_HW_CHANNEL_MAX (generated by the number of OcuHWSpecificSettings ticks) and this might be higher than the actual number of channels used in that configuration. <*>Preconditions: <*>Have extra channels ticked in OcuHwSpecificSettings than used OcuChannel in any variant <*>Test Case ID (internal TC that caught the defect) - optional <*>tse_bbx_wir_ocu_00100.c <*>Observed behavior: <*>Sometimes the code was entering an exception interrupt. <*>Expected behavior: <*>No issues. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Update the channel loop in Ocu_Ftm_DeInit from OCU_FTM_HW_CHANNEL_MAX to pFtmIpConfig->u8NumChannels. <*>pFtmIpConfig->u8NumChannels is specific for each variant.
MCAL-18126	Bug	[PWM] Fix reload point<*>Detailed description (how to reproduce it): <*>Reload

Table continues on the next page...

ID	Subtype	Headline and Description
		points is not enable in up count mode when options End cycle reload and Half cycle reload is not chosen. So End cycle reload must be always chosen when in up count mode. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18127	Bug	[PWM] ChannelEdgeSetup does not get data properly<*>Detailed description (how to reproduce it): <*>Pwm_PBCfg.c does not get data from ChannelEdgeSetup properly <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-18128	Bug	[CAN]Fix missing when Pretended Networking is enabled<*>Detailed description (how to reproduce it): <*>Calculate payload wrong. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>TS_ 91 <*>Observed behavior: <*>[...] <*>Expected behavior: <*>correctly Calcalute payload. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Reverse between byte high and byte low.
MCAL-18129	New	New Feature [MCEM] Implementation of software locking mechanism MCEM is a safety module, it is required to have locking mechanism to prevent incorrect sequence to initialize with more than 1 configuration.
MCAL-18130	New	New Feature [GPT] Update plugin_macros file for prescaler values computation New values used in xdm configuration, as deviders and not register value, needs updated support in plugin macros.
MCAL-18131	Bug	[ADC] Fix misra errors for S32K 4.2<*>Detailed description (how to reproduce it): <*>Please correct the misra errors are reported <*>Preconditions: <*>PVT_B53917_ADC_S32K_4.2_100_V03 <*>Test Case ID (internal TC that caught the defect) - optional <*>N/A <*>Observed behavior: <*>N/A <*>Expected behavior: <*>N/A <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>N/A

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18134	Bug	[PORT] Fix misra errors for S32K14x<*>Detailed description (how to reproduce it): <*>please see misra log file in attach <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...] ? Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18135	Bug	[ICU] Fix misra violation for S32K ASR 4.2<*>Problem detailed description (how to reproduce it): <*>In the current code, there are some misra violations. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>Misra violation <*>Expected behavior: <*>Misra violations were fixed or commented. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-18139	Bug	[OCU] S32K FTM Channels used in ISR generate not true<*>Detailed description (how to reproduce it): <*>Ocu driver does not have output. <*>OCU FTM Channels used in ISR can't generate... <*>Macros for channels used in ISR on output file have nothing... <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Output should be generate channels used in ISR..
MCAL-18140	New	New Feature [MCU] Fix Misra errors and Compiler warnings for S32K RTM 1.0.0 ASR 4.2.2 See attachment
MCAL-18142	Bug	[GPT] Update misra comments for S32K14x 4.2 files<*>Detailed description (how to reproduce it): <*>Please see attach station report and correct the misra errors. <*>Proposed solution (Optional): <*>Update the code to avoid misra warnings/error and update the misra comments where the code cannot be changed.
MCAL-18145	Bug	[OCU] Missing some resource for S32K14X 4.2<*>Detailed description (how to reproduce it): <*>some resource missing for all DERIVATIVE <*>FTM_2_CH_6 & FTM_2_CH_7 are missing... <*>Proposed solution (Optional): <*>FTM_2_CH_6 & FTM_2_CH_7 adding for all DERIVATIVE...

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18146	Bug	[CAN]Fix misra<*>Detailed description (how to reproduce it): <*>In misra report contains some missing. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Fix all missing which are found. <*>Note: in the ? Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix misra.
MCAL-18147	Bug	[SPI] Fix misra error<*>Detailed description (how to reproduce it): <*>Mcl_DmaGetChannelTcdAddress change prototype to return pointer instead of value. <*>So some misra errors appear. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Violates MISRA 2004 Rule 11.4 when use the function Mcl_DmaGetChannelTcdAddress <*>Expected behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>NA
MCAL-18150	Bug	[ETH] the same buffer is allocated twice before it is released<*>Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>In case user don't use transmission confirmation <*>* If the Eth_Transmit is interrupted and the Eth_ProvideTxBuffer is called when the Eth_ENET_u8SearchTxBufFrom = BufIdx which was passed for Eth_Transmit. In this case, the similar problem may occur since the same buffer is allocated twice before it is released. <*>e.g This problem could occur in following scenario with 4 TX buffers: TXBUF0, TXBUF1, TXBUF2, TXBUF3. <*># Allocate buffer TXBUF0 <*># Allocate buffer TXBUF1 <*># Allocate buffer TXBUF2 <*># Allocate buffer TXBUF3 (The Eth_ENET_u8SearchTxBufFrom[0] will get be wrapped to 0 here). <*># Eth_Transmit TXBUF1 <*># Eth_Transmit TXBUF2 <*># Eth_Transmit TXBUF3 <*># Eth_Transmit TXBUF0, the Eth_Transmit function get interrupted and Eth_ProvideTxBuffer is called after setting Eth_au8TxBufFlags[u8CtrlIdx][0] = ENET_TXB_LOCK_U8 | ENET_TXB_FIRST_U8 | ENET_TXB_LINK_U8; before setting TxBD[0]= (ENET_TXBD_R_U32 | ENET_TXBD_TO1_U32 | ENET_TXBD_L_U32 | ENET_TXBD_TC_U32) inside Eth_Enet_Transmit. <*>At this time, Eth_ENET_u8SearchTxBufFrom[0] = 0 and TxBD[0] & ENET_TXBD_R_U32 = 0, so the buffer TXBF0 is allocated again and Eth_au8TxBufFlags[u8CtrlIdx][0] is set again to ENET_TXB_LOCK_U8 | ENET_TXB_FIRST_U8. <*>9. Eth_Transmit TXBUF0. The buffer is transmit again <*>Expected behavior: <*>The buffer don't transmit again <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18151	Bug	[ICU] Fix wrong define user interrupt ASR 4.2<*>Detailed description (how to reproduce it): <*>User interrupt define need generate for all variant configuration (VS_0, VS_1, ...) <*>Preconditions: <*>[...] <*>Test Case ID

Table continues on the next page...

ID	Subtype	Headline and Description
		(internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>User interrupt define need generate for all variant configuration (VS_0, VS_1, ...) <*>... <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>using IcuHwInterruptConfigList
MCAL-18152	Bug	[ICU] The wrong wake up source is checked when EcuMWakeupSource symbolic name defines as EcuMWakeupSourceID<*>Problem detailed description (how to reproduce it): <*>The wakeup source is refered by the driver via symbolic name. <*>According to Autosar specification EcuM2166 and EcuM151_Conf, the EcuMWakeupSource. <*>[EcuM2166] The EcuMWakeupSourceId (see EcuM151_Conf) field in the EcuMWakeupSource container shall define the position corresponding to that wakeup source in all instances the EcuM_WakeupSourceType bitfield. <*>ECUM151_Conf <*>Name EcuMWakeupSourceId {WakeupSourceName} <*>Description: This parameter defines the identifier of this wakeup source. <*>Multiplicity 1 <*>Type EcucIntegerParamDef (Symbolic Name generated for this parameter) <*>Range 0 .. 31 <*>The symbolic name could be defined by the ID, in this case it would be the bit position but not bit field as defined in EcuM stub delivered in MCAL package. Therefore the expected wakeup source will not be set or will not be checked by from the driver. <*>CE's comment: Driver should calculate the bit field from EcuMWakeupSourceId value rather than using EcuMWakeupSource symbolic name. The similar issue in other drivers such as LIN, GPT <*>Fo Icu driver, in Icu_PluginMacros.m, replace line 500: <*>*/!]EcuMConf_EcuMWakeupSource_[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/@name"!][!/* <*>by: <*>*/!]1<<[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/EcuMWakeupSourceId"!][!/* <*>Preconditions: <*>The EcuM defines the EcuMWakeupSource as thevalue configured for EcuMWakeupSourceID. <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Trigger: <*>NA <*>Observed behavior: <*>The correct wake up source is not checked/set by the driver. <*>Expected behavior: <*>The correct wake up source is not checked/set by the driver. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional):
MCAL-18154	Bug	[PWM] Missing resource for S32K148 and channel limit is too small<*>Detailed description (how to reproduce it): <*>Missing resource PWM_FTM_5_CH_6_ISR for s32k148_lqfp144 <*>Channel FTM must follow resouce , not 32 channel <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...] ? Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18155	Bug	[ICU] Correct set input filter for FTM channel<*>Detailed description (how to reproduce it): <*>only set input filter for channel 0 <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Set input filter for channel 0, 1, 2, 3 <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>add shift for value input filter corresponding channel
MCAL-18156	Bug	[MCL] Correct set input filter for FTM channel<*>Detailed description (how to reproduce it): <*>only set input filter for channel 0 <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Set input filter for channel 0, 1, 2, 3 <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>add shift for value input filter corresponding channel
MCAL-18157	Bug	[MCEM] Soft Lock feature does not work correctty<*>Detailed description (how to reproduce it): <*>Soft Lock feature always set equal TRUE even using Unchecked in McemSoftLockedConfiguration, this issue make configuration be locked when call Mcem_init() at the first time and never configure again. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18160	Bug	[PWM] Fix misra for S32K14<*>Detailed description (how to reproduce it): Fix misra <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18161	Bug	[MCEM] Fix compiler warnings<*>Detailed description (how to reproduce it): <*>There are following compiler warnings in the current code: <*>C:/EB/tresos/23.0.0/plugins/Mcem_TS_T40D2M10I0R0/src/CDD_Mcem.c", line 239: warning #550-D: <*>variable "Mcem_pConfigPtr" was set but never used <*>static P2CONST(Mcem_ConfigType, AUTOMATIC, MCEM_APPL_CONST) Mcem_pConfigPtr = NULL_PTR; <*>"C:/EB/tresos/23.0.0/plugins/Mcem_TS_T40D2M10I0R0/src/Mcem_Eirm.c", line 219: warning #550-D: <*>variable "u32ReadInjectAddr" was set but never used <*>static uint32 u32ReadInjectAddr = 0U; <*>These warnings should be fixed or comment in the code. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected

Table continues on the next page...

ID	Subtype	Headline and Description
		behavior: <*>NA <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix or comment the compiler warnings
MCAL-18171	Bug	[PORT] The driver should support all Unused GPIO Pin Termination part 2<*>Imported from CQ ticket ENGR00390618. <*>Problem detailed description (how to reproduce it): <*>According to Application note AN5220, there ae some recommendation for "Unused GPIO Pin Termination" in page 30-"6.3 Unused GPIO Pin Termination" <*>http://www.nxp.com/assets/documents/data/en/application-notes/AN5220.pdf <*>For 176 or 100 pin device, the port driver does not support all implemented PAD yet. (0-148 for 176 pin device, 0-158 for 100 pin device). <*>According to this application note, the recommended configuration is (3) GPIO input with pull up/down or (4) GPIO output with pull up/down. <*>Therefore the port driver should support configure all implemented PADs (MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device). <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Trigger: <*>[...] <*>Observed behavior: <*>(MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device) are not configured by port driver. <*>Expected behavior: <*>(MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device) are configured by port driver. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>AN5220 chapter -"6.3 Unused GPIO Pin Termination" <*>Proposed solution (Optional): <*>[...]
MCAL-18172	Bug	[MCL] Inconsistent usage of function type<*>Detailed description (how to reproduce it): <*>return value type of function Mcl_IPW_DmaGetChannelTcdAddress() does not match the function type (at line 689 in Mcl_IPW.c file) <*>FUNC(P2VAR(Mcl_DmaTcdType, AUTOMATIC, MCL_APPL_DATA), MCL_CODE) Mcl_IPW_DmaGetChannelTcdAddress <*>(<*>P2CONST(Mcl_DmaConfigType, AUTOMATIC, MCL_APPL_CONST) Mcl_DmaConfigPtr, VAR (Mcl_ChannelType,AUTOMATIC) ChannelNumber <*>) <*>\{ <*>/* Return the TCD address for the hw channel corresponding to ChannelNumber */ <*>return (DMA_TCD((uint32)(*(Mcl_DmaConfigPtr->pChannelsConfig))[ChannelNumber].Dma_Channel)); <*>} <*>
MCAL-18173	Bug	[PWM] Wrong validate Pwm_Ftm_aNotifToChannelMap array in fucntion Pwm_Ftm_ValidateIdleState<*>Detailed description (how to reproduce it): <*>if (PWM_NO_EDGE != Pwm_Ftm_aNotifToChannelMap[u8channelid]) <*>\{ <*>nRetVal = (Std_ReturnType)E_NOT_OK; <*>break; <*>} <*>affter pwm_setoutputoidle() : Pwm_Ftm_aNotifToChannelMap[u8channelid]) =PWM_TOF_IRQ_FAULT_IDLE_STATE <*>so nRetVal is alway equal E_NOT_OK <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...] ? Detailed description (how to reproduce it): <*>[...] <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field,

Table continues on the next page...

ID	Subtype	Headline and Description
		please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18175	Bug	[MCEM] Wrong behavior of interupt routine: MCEM_EIRM_FAULT_SINGLEBIT_ISR and MCEM_EIRM_FAULT_DOUBLEBIT_ISR<*>Detailed description (how to reproduce it): <*>The Scenario is: <*>Fault ID 0 not using interrupt <*>Fault ID 2 using interrupt <*>then call Mcem_InjectFaults(Fault ID 0); Mcem_InjectFaults(Fault ID 2) the interrupt occur, but there is a problem with the interrupt routine: <*>the MCEM_EIRM_FAULT_SINGLEBIT_ISR or CEM_EIRM_FAULT_DOUBLEBIT_ISR still process the fault which is not enable interrupt with call callback function. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>[...] <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>[...]
MCAL-18178	Bug	[CAN]Fix missing in Trace report<*>Detailed description (how to reproduce it): <*>In trace report contains some missing in Trace Report. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Fix all missing. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix missing
MCAL-18179	Bug	[PWM] EnableNotification clear previous config of TOF<*>Detailed description (how to reproduce it): <*>TOF is used for fault related errata handling. In this case if channel is in Idle state, a software flag is asserted and TOF is enable to process fault in TOF interrupt. Pwm_Enablenotification clear this flag each time it is called. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>Needed TOF is not achieved <*>Expected behavior: <*>Needed TOF is achieved <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Only setting needed bit field of Pwm_Ftm_aNotifToChannelMap <*>call Pwm_EnableNotification in the case limited duty, set the flag PWM_TOF_IRQ_NO_EDGE_NOTIF
MCAL-18180	Bug	[GPT] Correct typo in configuration xdm on GptLptmrPrescaler and alternate<*>Detailed description (how to reproduce it): <*>GPT_MCAL_TMGptLptmrPrescaler and alternate have a typo on invalid checking; <*><a:tst expr="&lt;65536" false="Only the values 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536 are allowed"/> <*>"&lt;65536" should be "&lt;65537" or <= 65536 <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>"&lt;65536" should be "&lt;65537" or <= 65536

Table continues on the next page...

ID	Subtype	Headline and Description
MCAL-18184	Bug	[MCEM] Fix Misra violations<*>Detailed description (how to reproduce it): <*>There are some misra violations which prevent generating Misra report. <*>Preconditions: <*>NA <*>Test Case ID (internal TC that caught the defect) - optional <*>NA <*>Observed behavior: <*>NA <*>Expected behavior: <*>Misra report generated successfully <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix or comment the misra violation
MCAL-18185	Bug	[CAN]Fix compiler warning<*>Detailed description (how to reproduce it): <*>In compiler warning report contains some missing. <*>Preconditions: <*>[...] <*>Test Case ID (internal TC that caught the defect) - optional <*>[...] <*>Observed behavior: <*>[...] <*>Expected behavior: <*>Fix them. <*>Note: in the ?Expected behavior? field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) <*>Proposed solution (Optional): <*>Fix all compiler warning.
MCAL-18186	New	New Feature [I2C] Update developer tests' configuration files for S32K14X ASR 4.2 Problem: * There is currently a legacy "Mcu.xdm" which fails at plugin generation. Proposed Solution: * Replace old "Mcu.xdm" file with the current updated version.

4.4 BETA 0.9.0

ID	Subtype	Headline and Description
ENGR00369362	NewWork	[BASE] Add "callmode=far" for MemMap RAMCODE section for FLS driver NewWork Description: On GreenHills compiler, accesses to flash function placed in RAM, generate a linker error at relocation. Explicit pragma is needed in order to generate a far call. The issue is caused probably by the fact that the caller and callee are placed in the same file. Proposed solution (Optional): Add "#pragma ghs callmode=far" in MemMap.h FLS RAMCODE section, for GHS compiler: #ifdef FLS_START_SEC_RAMCODE ... #pragma ghs section text=".ramcode"

Table continues on the next page...

ID	Subtype	Headline and Description
		#pragma ghs inlineprologue #pragma ghs callmode=far Add "#pragma ghs callmode=default" at the end of the section, in order to revert to default behavior: #ifdef FLS_STOP_SEC_RAMCODE ... #pragma ghs section #pragma ghs noinlineprologue #pragma ghs callmode=default
ENGR00340892	Defect	[BASE] Add cache flush and invalidate macro defines for SW workaround Problem detailed description (how to reproduce it): On Halo platform there is no HW support to define non-cached sections. ETH driver need non-cached section to work. As a workaround we declare in Mcal.h file two macro defines for cahce_flush and cache_invalidate to be called by the Eth driver. Preconditions: Halo platform Test Case ID (internal TC that caught the defect) - optional Trigger: Configure system with caches. Observed behavior: No received/transmitted packets using Eth driver Expected behavior: Eth driver send and receive packets. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): As a workaround we declare in Mcal.h file two macro defines for cahce_flush and cache_invalidate to be called by the Eth driver.
ENGR00362004	NewWork	[BASE] Add support for different core architectures NewWork Description: Update Platform_Types.h to be able to support 32 bits and 64 bits cores Mcal_ARM.h - to be updated to support M, A32 and A64 cores In case two different cores have to be supported the Platform_Types.h and Mcal.h has to be generated. (The selection is done base on a RESOURCE parameter)
ENGR00357233	Defect	[BASE] Avoid multiple definition of STATIC macro Changes are required in Mcal.h CR description after analysis: === STATIC macro shoudl be removed from Mcal.h; this definition is not required by Autorsar 4.0 standard. All componenets should use the C keyword 'static' instead. Original CR description below: ===

Table continues on the next page...

ID	Subtype	Headline and Description
		Macro for keyword "STATIC" shall be defined with protection against multiple re-definition in order to avoid build warnings from other 3rd party code, i.e. change #define STATIC static by #ifndef STATIC #define STATIC static #endif
ENGR00367702	Defect	[BASE] Correct IAR warning messeges Correct IAR warning messages.
ENGR00331682	Defect	[BASE] Fls compilation error Problem detailed description (how to reproduce it): Customer have a compilation error in the FLS driver for calypso (0.9.0), at line 5005: PACKED P2VAR(Fls_LLD_DataBusWidthType, AUTOMATIC, FLS_APPL_DATA) dataPtrUnaligned = NULL_PTR; It seems to be due to the PACKED keyword. It is taking the defined one for windriver diab compiler (__packed__, which is correct) in Mcal.h. Here is the error message : ..\tools\wr\mpc5748_wr593\diab\5.9.3.0\WIN32\bin\dcc.exe -c -Xenum-is-best -Xrtti-off -Xexceptions-off -Xforce-declarations -ee1481 -tPPCVLEES:simple -g3 -XO -Xsize-opt -DTGT_MPC5748_WR593 -DFREESCALE_OS -DAUTOSAR_OS_USED -DOSDIABPPC -DOSDIABPPC -DOSDIABPPC -DOSDIABPPC -DADC_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DCAN_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DGPT_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DICU_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DLIN_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DPWM_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DSPI_INTERRUPT_TYPE=MCAL_ISR_TYPE_NONE -DOSDIABPPC -DOSDIABPPC -DTGT_DBG -IC:\CITECT\workspaces\ABC_NG\trunk\01_CODE\tools\wr\mpc5748_wr593\diab\5.9.3.0\include -Ilib\mtl -Ibsw\mcal\mcalAS\inc -Ibsw\mcal\biosAbcdCalypso\include -Ios\aos\inc -Ios\aos -I. -Ios\aos\inc -Ios\aos\conf bsw\mcal\mcalAS\src\Fls.c -o bsw\mcal\mcalAS\src\Fls.o "bsw\\mcal\\mcalAS\\src\\Fls.c", line 5005: error (dcc:1525): identifier Fls_LLD_DataBusWidthType not declared "bsw\\mcal\\mcalAS\\src\\Fls.c", line 5005: error (dcc:1086): redeclaration of Fls_LLD_DataBusWidthType "bsw\\mcal\\mcalAS\\src\\Fls.c", line 5005: error (dcc:1633): parse error near 'Fls_LLD_DataBusWidthType' "bsw\\mcal\\mcalAS\\src\\Fls.c", line 5005: error (dcc:1525): identifier dataPtrUnaligned not declared scons: *** [bsw\mcal\mcalAS\src\Fls.o] Error 1 Preconditions: Customer does not use -Xc-new compiler option which seems to cause the issue (if it is used then there is no issue). Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See attachment.
ENGR00353853	Defect	[BASE] The short name of Std_Types is Std but not STD_TYPES Problem detailed description (how to reproduce it): Std_Types.h file in BASE module will change its macros for version checking from STD_TYPES_AR_RELEASE_MAJOR_VERSION to STD_AR_RELEASE_MAJOR_VERSION . All other modules that include this file must adjust their version checking code for this accordingly. Customer is reporting issue in the integration with their WdgIf module 1. Within WdgIf module: Wrong AR version of the Standard Types used : The problem appears because WdgIf module uses "STD_AR_RELEASE_MAJOR_VERSION" to check the version of Std_Types, while Std_Types defines "STD_TYPES_AR_RELEASE_MAJOR_VERSION". According to AUTOSAR_TR_BSWModuleList document, the short name(API service prefix) of Std_Types.h is 'Std', witch will lead to STD_AR_RELEASE_MAJOR_VERSION being defined instead STD_TYPES_AR_RELEASE_MAJOR_VERSION.
ENGR00347013	NewWork	[BASE] Updates to support different ARM architectures NewWork Description: Updates to support different ARM architectures
ENGR00357978	Defect	[CAN] Driver have error ralate variable notification in template file Problem detailed description (how to reproduce it): the old code is imprecise: [!/* Minimum ratio between the peripheral clock frequency and Can Bit rate Check */!] [!VAR "maxmb" = "0"!] [!LOOP "../../CanHardwareObject/*"!] [!SELECT "node:ref(CanControllerRef)"!] [!IF "num:i(CanControllerId) = num:i($x)"!] [!VAR "maxmb" = "$maxmb+1"!] [!ENDIF!] [!ENDSELECT!] [!ENDLOOP!] This problem make don't check correctly the ratio between peripheral clock frequency and CAN bit rate. Proposed solution (Optional): Replate the old code by: [!VAR "maxmb" = "0"!] [!VAR "x" = "num:i(../../CanControllerId)"!] [!LOOP "../../../../CanHardwareObject/*"!] [!SELECT "node:ref(CanControllerRef)"!]

Table continues on the next page...

ID	Subtype	Headline and Description
		[!IF "num:i(CanControllerId) = num:i($x)"!] [!VAR "maxmb" = "$maxmb+1"!] [!ENDIF!] [!ENDSELECT!] [!ENDLOOP!]
ENGR00368170	NewWork	[CAN] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user. - When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or continute when event is set.
ENGR00368307	Defect	[CAN] Improve definition of Can_IdPtrType to prevent issue while using different definition for Can_IdType NewWork Description: In our simple demo Can application there is the following GHS compiler warning: simple_demo_can_rte\output\generated\src\Can_PBcfg.c, 414: warning 144: a value of type "const Can_IdType *" cannot be used to initialize an entity of type "Can_IdPtrType" Can_FilterMasks0_PB, With the Diab compiler a compiler error occurs: ".\simple_demo_can_rte\output\generated\src\Can_PBcfg.c", line 414: error (dcc:1552): initializer type `unsigned long const *' incompatible with object type `unsigned short const * const' In Can_PBcfg.c Can_FilterMask0_PB is defined as follow: static CONST(Can_IdType, CAN_CONST) Can_FilterMasks0_PB[CAN_MAXFILTERCOUNT_0] = { /* FilterMasks0[0], "AcceptAllExt" */ (Can_IdType)0x0U, /* FilterMasks0[1], "FilterMask_2047" */ (Can_IdType)0x7ffU }; The CanConfigSet variable which is of type Can_configType looks like: CONST(Can_ConfigType, CAN_CONST) CanConfigSet = { /* Number of CAN controllers configured */ (uint8) 1U, /* pFilterMasks */ Can_FilterMasks0_PB, /* MessageBufferConfigContainer */ { /* pMessageBufferConfigsPtr */ MessageBufferConfigs0_PB, /* uMessageBufferConfigCount */ (Can_HwHandleType)2U, }, /* FlexCAN controller description */ ControlerDescriptors0_PB,

Table continues on the next page...

ID	Subtype	Headline and Description

		#if (CAN_RXFIFO_ENABLE == STD_ON)
		/* Can_RxFiFoTableIdConfigType */
		NULL_PTR, /* Rx fifo disabled */
		#endif /* (CAN_RXFIFO_ENABLE == STD_ON) */
		/*Maximum Object IDs configured */
		(uint32)2U,
		/*Controller ID mapping*/
		{0U,0U},
		/*Can Object Type mapping*/
		{(Can_ObjType)CAN_RECEIVE
		,(Can_ObjType)CAN_TRANSMIT
		}
		};

		The second element of this set is of type Can_IdPtrType and so the compiler warning/error occurs.
		In Can.h Can_IdPtrType is defined:
		/*
		* @{
		* @brief Can_IdPtrType
		* @details Type for storing pointer to the Identifier Lenght Type.
		* - used by "Can_ConfigType" structure (pointer to the FilterMasks).
		*
		*/
		#if (CAN_EXTENDEDID == STD_ON)
		typedef CONSTP2CONST(uint32, CAN_CONST, CAN_APPL_CONST) Can_IdPtrType;
		#else /* (CAN_EXTENDEDID == STD_OFF) */
		typedef CONSTP2CONST(uint16, CAN_CONST, CAN_APPL_CONST) Can_IdPtrType;
		#endif /* (CAN_EXTENDEDID == STD_OFF) */
		/**@}*/
		Can_IdType is defined in Can_GeneralTypes.h file which can be uint16 or uint32.
		Should Can_IdPtrType only be defined in this way?
		typedef CONSTP2CONST(Can_IdType, CAN_CONST, CAN_APPL_CONST) Can_IdPtrType;
		Remark: this issue occur because customer using different definition for Can_IdType. Therefore, we are not able to detect this issue.
		Expected behavior:
		Requirement source:
		(e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
		Proposed solution (Optional):

ENGR00368582	Defect	[CAN] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers.

Table continues on the next page...

ID	Subtype	Headline and Description
		- Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00368810	Defect	[CAN] Limit the value for baudrate parameters related to FD Problem detailed description (how to reproduce it): When introduce FD feature, the implementation was based on ASR document which specified the value for the following parameters: CanControllerPropSeg: 0-255 CanControllerSeg1: 0-255 CanControllerSeg2: 0-255 CanControllerSyncJumpWidth: 0-255 However, there is limitation in our HW that we have only 3 bits for these values.

Table continues on the next page...

ID	Subtype	Headline and Description
		Therefore, we can not fully followed the ASR requirement there. AnalysisChamp will check whether we need raise CR to get deviation for these values. Preconditions: Enable FD Test Case ID (internal TC that caught the defect) - optional Trigger: Observed behavior: Expected behavior: Has error when configure incorrect values Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Limit the value and generate error when configure incorrect value
ENGR00367421	NewWork	[CAN] Support function Can_SetBaudrate in Can driver NewWork Description: When using FD, there will be 2 different baud rate (for data phase and control phase). The current APIs only support change baud rate with the baud rate input got from control phase. There will be problem when 2 baud rate config with the same baud rate for control phase but different in data baud rate. In order to let customer change the baud rate more flexible when using FD mode, we will implement requirement SWS_CAN_00491 from ASR 4.2.1. Then the we have api Can_SetBaudrate. Expected behavior: Able to use index of config to change baud rate Requirement source: NA for ASR 4.0.3. We will discuss to have PRD requirement later (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Merge implementation from ASR 4.2.1
ENGR00368953	Defect	[CAN] Wrong in error message mentioned on the range of FdTimeSegment2 Problem detailed description (how to reproduce it): Line 638 in Can_PBCfg.c: [!IF "($FdTimeSegment2<2) or ($FdTimeSegment2>8) "!] [!ERROR!] FdTimeSegment1 should have values between 2 and 8 [!ENDERROR!] [!ENDIF!] Check on the range of FdTimeSegment2, but the error message mentions FdTimeSegment1 This check can however be removed completely since it is redundant with the ValueRange of CanControllerSeg2 in Can.xdm.
ENGR00368162	NewWork	[DIO] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user.

Table continues on the next page...

ID	Subtype	Headline and Description
		- When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or contitute when event is set.
ENGR00368574	Defect	[DIO] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl

Table continues on the next page...

ID	Subtype	Headline and Description
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00369257	NewWork	[FLS] Add section attribute for AccessCode function on Linaro(GCC) compiler NewWork Description: Because the Linaro(GCC) compiler does not support defining block sections for code placement, place the AccessCode function(Fls_Flash_AccessCode) in a dedicated section using: __attribute__ ((section (".acfls_code_rom"))) , on the function declaration. For example, change: void Fls_Flash_AccessCode(P2VAR(volatile uint8, AUTOMATIC, FLS_CODE) regBasePtr, P2FUNC(void, FLS_CODE, CallBack)(void)); to: #ifdef __GNUC__ void Fls_Flash_AccessCode(P2VAR(volatile uint8, AUTOMATIC, FLS_CODE) regBasePtr, P2FUNC(void, FLS_CODE, CallBack)(void)) __attribute__ ((section (".acfls_code_rom"))) ; #else void Fls_Flash_AccessCode(P2VAR(volatile uint8, AUTOMATIC, FLS_CODE) regBasePtr, P2FUNC(void, FLS_CODE, CallBack)(void)); #endif
ENGR00368351	Defect	[FLS] FLS-Driver causes code execution from RAM The FLS-Driver contains one function. * Fls_Flash_AccessCode) ... which requires code execution from RAM. This feature does not work under all conditions because it's blocked by HW. This function is only able with exclusions. Which are: The FLS driver is not allowed to write or erase a sector which resides in the same read-while-write partition as the sector from which the flash driver access code is being executed. In addition the function must be mapped to be located in the flash. (This reflects just my understanding of the matter and could be imprecise.) Requests: * Please define in a way that is easily understandable by everyone (incl. Customers) how to map, link, and call this function in order be working in general. * In addition please take actions that this function cannot erase the flash block in which itself resides.
ENGR00368167	NewWork	[FLS] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user. - When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or continute when event is set.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00368575	Defect	[GPT] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00297387	NewWork	[GPT] Offer support for changing the base clock of the controlled hardware (PR-MCAL-3196) NewWork Description: Implement PR-MCAL-3196 The Gpt driver shall provide an optional API and configuration parameters for changing the base clock of the controlled hardware. Per default this optional functionality and configuration parameters shall be disabled. Requirement source: CPRT - BLN_CPRT_SMCAL_4.0_01.00.00
ENGR00367551	NewWork	[I2C] Add DMA support NewWork Description: Add DMA support for the master communication Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00367572	NewWork	[I2C] Add slave support NewWork Description: Add slave support Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00369470	NewWork	[I2C] Copy the Java files from Gpt to I2C NewWork Description: The files from the Java folder in Gpt should be moved to Base because the I2C driver also uses the noderef function defined in the JAR file. Proposed solution (Optional): 1. Move the Java folder from Gpt to the Base generic folder 2. Add in the makefile the following: ENABLE_COPY_TO_TRESOS_BEFORE_PLUGIN_GENERATION=ON SRC_FILES_TO_COPY_BEFORE_PLUGIN_GENERATION= $ (MODULE_PATH)/generic/Java/ com.freescale.tools.tresos.xpath.jar@bin=y,outdir=..,instdir=..

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00368571	Defect	[I2C] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00361342	Defect	[ICU] Avoid multiple definition of STATIC macro Changes are required in Mcal.h CR description after analysis: == STATIC macro shoudl be removed from Mcal.h; this definition is not required by Autorsar 4.0 standard. All componenets should use the C keyword 'static' instead. Original CR description below: == Macro for keyword "STATIC" shall be defined with protection against multiple re-definition in order to avoid build warnings from other 3rd party code, i.e. change #define STATIC static by #ifndef STATIC #define STATIC static #endif

ENGR00364915	Defect	[ICU] Fix the define number of Ftm channels and Ftm Modules
		
		Problem detailed description (how to reproduce it):
		Now on Ftm_Common.c the definition of Ftm channels and Ftm Modules always is 8 and 4, but on treerunner platform the number of channels and modules is 6 and 2. In somewhere in code, used define instead of the hard code.
		For example:
		In function Icu_Ftm_DeInit:
		for (u8hwModuleNo = 0U ; u8hwModuleNo < 4U ; u8hwModuleNo++)
		{
		u8srcClk = (uint8)((u32GlobalConfig & ICU_FTM_SRC_CLK_MASK_U32) >> ICU_FTM_SRC_CLK_SHIFT);
		u32GlobalConfig = u32GlobalConfig >> ICU_FTM_GLOBAL_CONFIG_WIDTH;
		if (u8srcClk != (uint8)0U)
		{
		Icu_Ftm_GlobalConfiguration((uint8)u8hwModuleNo, (uint8)0, (uint8)0);
		}
		}
		should used the define for compare with u8hwModuleNo.
		Proposed solution (Optional):
		The number of channel and module should be defined on <Mod>_Cfg.h via information on resource file

ENGR00365669	Defect	[ICU] Incorrect version checking in configuration template files Please update the following files in specific/generate folder of the driver: - [DRV]_VersionCheck_Inc.m - [DRV]_VersionCheck_Src.m - [DRV]_VersionCheck_Src_PB.m in order to implement all of the following topics that apply : 1. Replace: - ArMajorVersion with ArReleaseMajorVersion - ArMinorVersion with ArReleaseMinorVersion - ArPatchVersion with ArReleaseRevisionVersion

Table continues on the next page...

ID	Subtype	Headline and Description
		2. Uncomment the part of the file that is performing version checking on the following variables (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion) 3. Add test to variable M4_SRC_SW_VERSION_PATCH against SwPatchVersion variable from xdm, in the files where this check is missing 4. As per requirement PR-MCAL-2674, replace the ASSERT statements with IF-ERROR statements Please see attached a corrected version for the file Port_VersionCheck_Inc.m
ENGR00351901	Defect	[ICU] Separation of Logical/Physical elements in driver to allow configuration of the Master Bus Clock Problem detailed description (how to reproduce it): Scenario: ICU channel: Emios0 ch11 Counter bus C (for the above ch11): Emios0 ch8 PWM measured by ICU: external signal The counter bus is configured in PWM module (PwmeMiosMasterBus). The problem in this case is that the counter bus is not set/configured although it's configured in PWM. Apparently the PWM will not set up the counter bus if it is not used by one of the configured PWM/EMIOS channels. At the moment the customer is using as workaround a dummy PWM channel, in order to get the Counter Bus C working for the mentioned ICU channel. Preconditions: see above. Test Case ID (internal TC that caught the defect) - optional [...] Trigger: N/A Observed behavior: Counter Bus C is not started. Expected behavior: Configure and start the counter bus if needed by the ICU channel. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00365981	NewWork	[ICU] Update after separating the logical and physical configurations NewWork Description: The following updates need to be done after CR:ENGR00351901 - Icu_Ipw_InitChannel should be named Icu_Ipw_Init (similar to the HLD function) Expected behavior: The driver should be updated after CR:ENGR00351901 Requirement source: Internal refactoring (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):
ENGR00368463	NewWork	[ICU] Update memMap section for local variables

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: Update memMap section for local variables. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update memMap section for local variables: #define ICU_START_SEC_VAR_INIT_UNSPECIFIED /** * @violates @ref Icu_c_REF_1 MISRA 2004 Required Rule 19.1 , only preprocessor statements * and comments before '#include' * @violates @ref Icu_c_REF_2 MISRA 2004 Advisory Rule 19.15, precautions to prevent the contents of * a header file being included twice. */ #include "MemMap.h" #if ((ICU_VALIDATE_GLOBAL_CALL == STD_ON) || (ICU_VALIDATE_CALL_AND_CHANNEL == STD_ON)) static VAR(Icu_eGlobalStateType, ICU_VAR) Icu_GlobalState = ICU_STATE_UNINIT; #endif /* ((ICU_VALIDATE_GLOBAL_CALL == STD_ON) || (ICU_VALIDATE_CALL_AND_CHANNEL == STD_ON)) */ #define ICU_STOP_SEC_VAR_INIT_UNSPECIFIED /* * @violates @ref Icu_c_REF_1 Violates MISRA 2004 Advisory Rule 19.1, only preprocessor statements * and comments before "#include" * * @violates @ref Icu_c_REF_2 precautions to prevent the contents * of a header file being included twice */ #include "MemMap.h"
ENGR00364239	NewWork	[ICU] Version checking is missing in configuration template files Problem detailed description (how to reproduce it): Some modules' configuration templates do not check if the version number in the configuration (stored in CommonPublishedInformation) matches the one hard-coded into the templates. These modules are: DIODriver, FlashDriver, FlashEEPROMEmulation, GPTDriver, ICUDriver, MCUDriver. CE comments: Customer would like to add somethings like this: [!VAR "ADC_AR_RELEASE_MAJOR_VERSION_TEMPLATE"="M4_SRC_AR_SPEC_VERSION_MAJOR"!][!// [!VAR "ADC_AR_RELEASE_MINOR_VERSION_TEMPLATE"="M4_SRC_AR_SPEC_VERSION_MINOR"!][!// [!VAR "ADC_AR_RELEASE_REVISION_VERSION_TEMPLATE"="M4_SRC_AR_SPEC_VERSION_PATCH"!][!//

Table continues on the next page...

ID	Subtype	Headline and Description
		[!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJOR"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINOR"!][!// [!ENDNOCODE!][!// [!SELECT "CommonPublishedInformation"!][!// [!/* [!ASSERT "ArReleaseMajorVersion = num:i($ADC_AR_RELEASE_MAJOR_VERSION_TEMPLATE)"!] **** AUTOSAR release major version number of the Basic Software Module Description file (Adc.epd version [!"ArReleaseMajorVersion"!]) and the Code template file (Adc_Cfg.c version [!"num:i($ADC_AR_RELEASE_MAJOR_VERSION_TEMPLATE)"!]) are different **** [!ENDASSERT!][!// Preconditions: BLN_SMCAL_4.0_PANTHER_RTM_1.0.0 Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please see Flexray, Adc driver, they are using *_VersionCheck.m. after that include this macro into template files
ENGR00366997	Defect	[CAN] Hardware registers CANx_RXIMRs are written incorrectly Problem detailed description (how to reproduce it): Configure FlexCAN_B with: CanControllerRxFifoEnable is true CanRxFifoFiltersNumber is FILTERS_NUMBER_24 Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional Can_TC_1020 in Can_TS_65 Trigger: Execution Observed behavior: The program jump to exception when Can_Init is called Expected behavior: The program does not jump to exception when Can_Init is called Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Root cause: The CANx_RXIMRs register are written incorrectly. When configure CanRxFifoFiltersNumber with FILTERS_NUMBER_24. The registers CANx_RXIMR0 - CANx_RXIMR11 are used to filter for FIFO (please refer the CANx_CTRL2). But In Can_FlexCan_ChangeBaudrate: The register CANx_RXIMR0 - CANx_RXIMR23 are written

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution: It need check the u8RxFifoTableId before write
ENGR00369102	Defect	[CAN] Incorrect processing when FIFO overflow occurred Problem detailed description (how to reproduce it): When the FIFO get overflow, the driver is stuck in function Can_FlexCan_ProcessRx and does not process any message. Therefore, driver has issue with the following configuration: - several PDUs mappped to HOH0 - CanControllerRxFifoEnable enabled - Filter allows all messages to pass - CAN 1ms polling Mode They experience FIFO overflow - although they get only 6 massages per millisecond. After an overflow FIFO stays in this state even after there are no messages on the bus. CE' comments: Fifo is enabled and overflow occurred: At the first time, if we have overflow on bus or warning (5MBs). Can driver was processing: 1. u8MbNodata = (uint8)1U; /*that mean do nothing in Can_FlexCan_ProcessRx()*/ 2. put MB index = last MB index and break out Det_ReportError(). Overflow will be keep in next transmit/receive cycle. The fixing which should contain following: - CAN FIFO should be polled (read out) up to the point when is becomes empty. - If there is a Warning Flag set then a Warning Notification should be triggered -> Warning Flag should be cleared under Fifo should be read out completely. - In case of overflow, an Overflow Notification should be called -> Overflow Flag should be cleared und der Fifo should be read out completely.
ENGR00367418	NewWork	[CAN] Update FD feature with support for tranceiver compensation NewWork Description: Update FD feature with support for tranceiver compensation Expected behavior: The value for TRCV it is generated from tresos in structure "Can_ControllerFdConfigType" with the following name "u32CanControllerTrcvDelayCompensation", but it is not used in driver code. The register field corresponding to TRCV delay should be initialized with values generated from TRESOS. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00367550	Defect	[CAN] Using improper solution to configure bits in register Problem detailed description (how to reproduce it): Please check again the following code:

Table continues on the next page...

ID	Subtype	Headline and Description
		in file Can_FlexCan.c line 1259 and many other positions: /*Clear FLEXCAN_FDCTRL*/ /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ REG_BIT_CLEAR32(FLEXCAN_FDCTRL(u8HwOffset), FLEXCAN_FDCTRL_MBDSR2_U32); /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ REG_BIT_SET32(FLEXCAN_FDCTRL(u8HwOffset), (uint32)0x0U<<FLEXCAN_MBDSR2_OFFSET); There is another way to use REG_RMW32 to replace for both 2 command. In this way, we will do this in a single command for a specific purpose. Proposed solution (Optional): Using REG_RMW32 instead of a couple of REG_BIT_CLEAR32 and REG_BIT_SET32
ENGR00368607	Defect	[PWM] The TOF and TOIE bits of SC Register are on other possitions compared with platforms Problem detailed description (how to reproduce it): On TreeRunner, Halo, Rayleight, the TOF and TOIE are put on 6, 7 position, but on S32K, these bits put on 8,9 position. Proposed solution (Optional): We should add the definition for IPV version on Soc_Ips.h, and then on IPV_FTM, we will check the IPV version, if the version is S32K, these bit need re-define on new position
ENGR00368524	Defect	[ICU] Change the sequence of configuration in enable mode functions Problem detailed description (how to reproduce it): The sequence of configuration is not logical, sometime we can not managed the unexpected interrupts when modes (edge detection, edge count and timestamp) were enabled. Preconditions: Run test and incorrect result. Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Should enable interrupt after finish configuring every that need for channel: For example in function Icu_Ftm_EnableEdgeDetection: Change from: /* Enable interrupts on the Ftm channel */ /* @violates @ref Icu_Ftm_c_REF_4 Violates MISRA 2004 Rule 11.1, Cast from unsigned long * to pointer.*/ REG_BIT_SET32(FTM_CSC_ADDR32(u8ModuleIdx,u8ChannelIdx),

Table continues on the next page...

ID	Subtype	Headline and Description
		FTM_CSC_CHIE_MASK_U32); Icu_Ftm_StartChannel(hwChannel); /* Set Edge Detect mode for the Ftm channel in the configuration array */ Icu_Ftm_SetChConfig(hwChannel, \ (Icu_Ftm_ChConfigType) ((Icu_Ftm_ChConfigType)FTM_CHANNEL_MODE_SIGNAL_EDGE_DETECT\ << FTM_ICU_MEAS_MODE_SHIFT)); To: Icu_Ftm_StartChannel(hwChannel); /* Set Edge Detect mode for the Ftm channel in the configuration array */ Icu_Ftm_SetChConfig(hwChannel, \ (Icu_Ftm_ChConfigType) ((Icu_Ftm_ChConfigType)FTM_CHANNEL_MODE_SIGNAL_EDGE_DETECT\ << FTM_ICU_MEAS_MODE_SHIFT)); /* Enable interrupts on the Ftm channel */ /* @violates @ref Icu_Ftm_c_REF_4 Violates MISRA 2004 Rule 11.1, Cast from unsigned long * to pointer.*/ REG_BIT_SET32(FTM_CSC_ADDR32(u8ModuleIdx,u8ChannelIdx), FTM_CSC_CHIE_MASK_U32);
ENGR00368939	Defect	[ICU] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section:

Table continues on the next page...

ID	Subtype	Headline and Description
		#define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00368999	Defect	[PWM] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00368144	Defect	[PWM] correct the issue related to channel-output disable feature Problem detailed description (how to reproduce it): Status And Control (FTMx_SC) register supports PWMENx bits which enables the PWM channel output. These bits do not set 1, therefore there is no pulse can be generate
ENGR00367522	Defect	[I2C] I2C_pDemCfgPtr used but not defined Problem detailed description (how to reproduce it): I2C_pDemCfgPtr is used but not defined. #if (I2C_DISABLE_DEM_REPORT_ERROR_STATUS == STD_OFF) extern P2CONST(I2C_DemConfigType, I2C_VAR, I2C_APPL_CONST) I2C_pDemCfgPtr; #endif must be added in I2C_LPI2C.c Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00369050	NewWork	[GPT][ICU][MCL] LPIT_BASE_ADDR32 variable should be moved to MCL Problem detailed description (how to reproduce it): IPV_LPIT are used on both GPT and MCL, so the LPIT_BASE_ADDR32

Table continues on the next page...

ID	Subtype	Headline and Description
		variable should be moved to MCL Proposed solution (Optional): - IPV_LPIT : Create new file Lpit_Common.c to declare the LPIT_BASE_ADDR32, and then this variable is externed on Reg_eSys_Lpit.h, by this way, Icu and Gpt both can use this variable - MCL : Please add Lpit_Common.c and Reg_eSys_Lpit.h on Mcl.mak file -GPT and ICU : Please remove Reg_eSys_Lpit.h on Gpt.mak and Icu.Mak
ENGR00369169	Defect	[SPI] Exception is generated when SPI driver was not initialized and an LPSPI interrupt occurred Problem detailed description (how to reproduce it): we run the test case with condition as: driver not initialized and the interrupt occurred, as expected: driver can run normally and the interrupt flag will be cleared. But with current implementation, the variable: Lpspi_Dev was not configured so the pointer of SR register address was NULL,if we try to modify it (clear status) the exception will occur. [...] Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00367488	Defect	[SPI] The SPI sequence will not end in Async mode Problem detailed description (how to reproduce it): In CS continue mode, the SPI sequence will not end in Async mode when use DMA or non-DMA transfers or data width 16 bit. LPSPI HW does not push the last received word to the RX FIFO until the next word is started or the transfer is ended. So, we cannot read last received word after each transfer word complete. [...] Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional SPI_TS_026 SPI_TS_005(Use DMA transfers) SPI_TS_006(Use DMA transfers) [...] Trigger: [...] Observed behavior: [...] Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00369039	Defect	[MCU] Fix Misra findings Problem detailed description (how to reproduce it): File "..\..\output\S32K14X_S32K144\mcu\Mcu_TS_M06_cfg10\generate\include\Mcu_Cfg.h", line 532, MISRA Rule Violated 8.7 (Required): no MISRA violation comment was found (maybe wrong format is used). File "c:\EB\tresos\14.2.1_kynetis\plugins\Mcu_TS_T40D2M9I0R0\src\Mcu.c", line 319, MISRA Rule Violated 8.10 (Required): no MISRA violation comment was found (maybe wrong format is used). File "c:\EB\tresos\14.2.1_kynetis\plugins\Mcu_TS_T40D2M9I0R0\src\Mcu_PCC.c", line 212, MISRA Rule Violated 8.10 (Required): no MISRA violation comment was found (maybe wrong format is used). Preconditions: MCU_INIT_CLOCK = STD_OFF Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00367545	NewWork	[MCU] Update the driver for S32K BETA NewWork Description: Update the Mcu driver for S32K BETA
ENGR00368168	NewWork	[MCL] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user. - When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or contitute when event is set.
ENGR00368546	Defect	[MCL]Correct the Macro in function Mcl_IPW_DmaClearDone Problem detailed description (how to reproduce it): In the function using the macro - REG_WRITE8((uint32)(DMA_MOD_BASE_CH_ADDR32((uint32)

Table continues on the next page...

ID	Subtype	Headline and Description
		(Mcl_DmaConfigPtr->pChannelsConfig[nChannel].Dma_Channel)) + DMA_CDNE_OFFSET_U32) , \ ((Mcl_DmaConfigPtr->pChannelsConfig[nChannel].Dma_Channel) & DMA_CTRL_MAX_CHANNELS_MASK_U8)) to clear the CDNE register. The issue is from "#define DMA_CDNE_OFFSET_U32 0x1F" in Reg_eSys_Dma.h. But in fact, the offset is difference depend on platform (in Panther RM is 0x1F, in S32K RM is 0x1C". So, if using the below macro to clear the CDNE register will occur the issue is the register can't be cleared. Proposed solution (Optional): Replace the macro - REG_WRITE8((uint32)(DMA_MOD_BASE_CH_ADDR32((uint32) (Mcl_DmaConfigPtr->pChannelsConfig[nChannel].Dma_Channel)) + DMA_CDNE_OFFSET_U32) , \ ((Mcl_DmaConfigPtr->pChannelsConfig[nChannel].Dma_Channel) & DMA_CTRL_MAX_CHANNELS_MASK_U8)) By REG_WRITE8((uint32)DMA_CDNE_ADDR32(Mcl_DmaConfigPtr- >pChannelsConfig[nChannel].Dma_Channel) , \ ((Mcl_DmaConfigPtr->pChannelsConfig[nChannel].Dma_Channel) & DMA_CTRL_MAX_CHANNELS_MASK_U8));
ENGR00367278	Defect	[MCL] Correct compiler warnings Problem detailed description (how to reproduce it): On S32K 0.8.0 has 2 compiler warnings and they will be fixed in BETA 0.9.0
ENGR00368157	NewWork	[MCU] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user. - When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or continute when event is set.
ENGR00368605	Defect	[MCU] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers. - Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to

Table continues on the next page...

ID	Subtype	Headline and Description
		define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00367279	Defect	[MCU] Remove the unused variable "Mcu_pClockConfig" Variable "Mcu_pClockConfig" was set but never used so we should remove it from "Mcu_IPW.c
ENGR00365449	Defect	[MCU] Xdm scheme contains duplicated attributes Problem detailed description (how to reproduce it): During reading of the Mcu.xdm file there are following warnings from Tresos studio: Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuFIRC/McuFIRC_Div defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuFIRC/McuFIRC_DivOutputValue defines two attributes with the same name "DEFAULT" which could not be merged.

Table continues on the next page...

ID	Subtype	Headline and Description
		Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuSIRC/McuSIRC_Div defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuSIRC/McuSIRC_DivOutputValue defines two attributes with the same name "DEFAULT" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuSXOSC/McuSXOSC_EOCV defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuSXOSC/McuSXOSC_Div defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuSXOSC/McuSXOSC_DivOutputValue defines two attributes with the same name "DEFAULT" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuFXOSC/McuFXOSC_EOCV defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuFXOSC/McuFXOSC_Div defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu/McuModuleConfiguration/McuClockSettingConfig/McuFXOSC/McuFXOSC_DivOutputValue defines two attributes with the same name "DEFAULT" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu defines two attributes with the same name "DESC" which could not be merged. Removing one of the attributes Node /TS_T2D35M10I1R0/Mcu defines two attributes with the same name "LOWER-MULTIPLICITY" which could not be merged. Removing one of the attributes
		Node /TS_T2D35M10I1R0/Mcu defines two attributes with the same name "UPPER-MULTIPLICITY" which could not be merged. Removing one of the attributes Proposed solution (Optional): Remove the redundant ones.
ENGR00368573	Defect	[PORT] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Compiler Abstraction Keywords and Memory Sections in our MCAL drivers. The main Issues identified are: Compiler Abstraction: - Inconsistence usage of Compiler Abstraction Keywords along MCAL drivers.

Table continues on the next page...

ID	Subtype	Headline and Description
		- Incorrect usage of Compiler Abstraction Keywords in MCAL drivers Memory mapping: - Variable/Const assignment to wrong memory sections. The AUTOSAR reference specifications we have to align to - Specification of Compiler Abstraction - V3.2.0 R4.0 Rev 3 - Specification of Memory Mapping - V1.4.0 R4.0 Rev 3 The slides attached tailor the AutoSAR specification for our MCAL4.0 in term of Memory Allocation Keywords as well as the Compiler Abstraction and try to define some rules to be follow by all developers for all drivers Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler Errors Expected behavior: NA Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): See the attachment Examples of what to correct: Memory sections not closed in the same file (or nested with other sections). Constants placed in VAR section: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl extern CONST(Mcu_ConfigType, MCU_CONST) Mcu_Config_PC; Uninitialized variables in VAR_INIT sections: #define MCU_START_SEC_VAR_INIT_UNSPECIFIED ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl #include "Mcu_MemMap.h" ',`dnl #include "MemMap.h" ')dnl
		static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];
ENGR00369180	Defect	[PORT] Update Invalid field for xdm file Problem detailed description (how to reproduce it): Invalid field of PortPinPcr, PortPinId, PortNumberOfPortPins is not correct Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00368156	Defect	[PWM] correct the issue related to channel-output disable feature Problem detailed description (how to reproduce it): Status And Control (FTMx_SC) register supports PWMENx bits which enables the PWM channel output. These bits do not set 1, therefore there is no pulse can be generate
ENGR00368694	Defect	[PWM]Correct fault callback table Problem detailed description (how to reproduce it): the order of fault callback in pfFaultNotification is incorrect was found in Pwm_Cfg.c and Pwm_PBCfg.c { [!"$FtmFault3Notif"!], [!"$FtmFault2Notif"!], [!"$FtmFault1Notif"!], [!"$FtmFault0Notif"!] } The valid order is below: { [!"$FtmFault0Notif"!], [!"$FtmFault1Notif"!], [!"$FtmFault2Notif"!], [!"$FtmFault3Notif"!] }
ENGR00369074	Defect	[PWM]Fix misra and compiler waring Problem detailed description (how to reproduce it): Fix misra violation and compiler warning reported by tester
ENGR00369450	NewWork	[SPI] Exception is generated when SPI driver was not initialized and a LPSPI interrupt occurred NewWork Description: we run the test case with condition as: driver not initialized and the interrupt occurred, as expected: driver can run normally and the interrupt flag will be cleared. But with current implementation, the variable: Lpspi_Dev was not configured so the pointer of SR register address was NULL,if we try to modify it (clear status) the exception will occur. Related with CR ENGR00369169 [...] Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00368159	NewWork	[SPI] Improve Dem/Det error description in Usermanual NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that should make clear with user. - When these events is set, condition to set event? - How to impact to system or something else if event is set... - Suggest user must be reaction or continute when event is set.
ENGR00344789	Defect	[SPI] Some local variables are wrongly guarded with SPI_DEV_ERROR_DETECT Problem detailed description (how to reproduce it): The multi configuration test (Spi_TS_Eq_Cot_01, Spi_TS_Define_Compile) will build fail with the changes of driver. Some local variables in Spi_SyncTransmit() function will be undefined when SPI_DEV_ERROR_DETECT=STD_OFF. Otherwise, Misra report has some errors as the same root cause. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00367245	Defect	[SPI] Template files were generated incorrect Problem detailed description (how to reproduce it): The test will build failed with the new driver plugin. The external device attributes were generated incorrect. Please see the test report in attachment. Test Case ID (internal TC that caught the defect) - optional Spi_TS_Eq_Cot_01 Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00367247	Defect	[SPI] The SPI sequence will not end in Async mode

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): The SPI sequence will not end in Async mode. This affects DMA and non-DMA transfers. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional SPI_TS_026 Trigger: [...] Observed behavior: [...] Expected behavior: [...] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00348487	NewWork	[SPI] Update info about Slave baudrate Curently, the driver supports upto 16Mhz baudrate for master mode and slave mode. However, in the slave receive only mode, the baudrate is upto 60Mhz (as stated in Datasheet The customer configures SPI baudrate is 40Mhz then EB Tresos throw an error message showing that the maximum baudrate is 16Mhz. Could SPI driver support upto 60Mhz baudrate for slave mode? As my understanding, there is no special setting for slave readonly mode excepting baudrate (it is upto 60Mhz). Therefore, only update the check of baudrate parameter in xdm file could be ok.
ENGR00367328	NewWork	[WDG] Add constraints in configuration to validate the inputs NewWork Description: Add constraint in configuration for window and timeout. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]"

4.5 EAR 0.8.0

This is the first release for the S32K14X MCAL 4.2 product.

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.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number Release Notes for S32K14X
AUTOSAR 4.2 SMCAL RTM 1.0.2
Revision 1.0