Release Notes

for MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5

Document Number: Release Notes for MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5
Rev. 1.0

Contents

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

Chapter 1
Getting Started

Chapter 2
Release Specifics

Chapter 3
Known Issues for MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5

Chapter 4
Changes List for MPC574XG AUTOSAR 4.0 MCAL

Chapter 1
Getting Started

1.1 Package content

This package contains the NXP MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5:

- "eclipse/plugins/<mod>_TS_T2D35M10I5R0" directories - Tresos Plugins, 1 per module.
- "MPC574XG_MCAL4.0_RTM_1.0.5_Sample_Application" - Folder containing the MCAL Sample Application.
- "MPC574XG_MCAL4.0_RTM_1.0.5_ReleaseNotes.pdf" - This file.
- "MPC574XG_MCAL4.0_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 MCAL 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_T2D35M10I5R0" directories and .JAR files to the "<Tresos Install Path> \plugins" directory.

Chapter 2
Release Specifics

The MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5 is AUTOSAR 4.0 Rev0003 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 4.0 MCAL RTM 1.0.5 release for the MPC574XG platform.

The Sample Application included in this release contains some basic examples of usage for the MCAL drivers. It also includes an example of integration of the MCAL drivers with the Freescale AUTOSAR OS/MPC574xG v4.0 RTM v1.0.5. (the OS installer comes as a different NXP software package).

The ETH driver included in this release is compliant with AUTOSAR R4.1 Rev 1.

The Can driver implements the CAN-FD extension as defined by ASR4.2.1

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 MCAL driver (*Deviation from Requirements* chapter).

This release was developed and tested using:

- EVB Motherboard X-MPC574XG-MB
- MPC574XG MINI-MODULE 256MAPBGA MPC574XG-256DS
- MPC574XG MINI-MODULE 324PBGA XMPC574XG324DSC2

- MPC574XG MINI-MODULE 324PBGA X-MPC574XG-324DS
- MPC574XG MINI-MODULE 176LQFP MPC574XG-176DS
- MPC574XG_6M DEVICE 256MAPBGA PPC5748GSAMMJ6 0N78S
- MPC574XG_6M DEVICE 324MAPBGA PPC5748GSAMMN6 0N78S
- MPC574XG_6M DEVICE 256MAPBGA PPC5748GK1MMJ6A 1N81M
- MPC574XC_3M DEVICE 176LQFP PPC5746CSMKU6 1N84S

The functions contained in the CanIf, Dem, Det, EcuM, EthIf, FrIf, LinIf, MemIf, 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
MPC5746C Reference Manual	Rev. 5, 10/2017
MPC5748G Reference Manual	Rev. 6, 10/2017

Table 2-2. Implemented Errata

Document Title	Maskset	Date
MPC5748G_1N81M_Rev. 3	1N81M	Nov-17
MPC5748G_0N78S_Rev. 2	0N78S	Nov-17
MPC5746C_1N84S_Rev. 2	1N84S	Nov-17

2.3 Supported Derivatives

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

- MPC5748G_LQFP176

- MPC5748G_MAPBGA256
- MPC5748G_MAPBGA324
- MPC5747G_LQFP176
- MPC5747G_MAPBGA256
- MPC5747G_MAPBGA324
- MPC5746G_LQFP176
- MPC5746G_MAPBGA256
- MPC5746G_MAPBGA324
- MPC5748C_LQFP176
- MPC5748C_MAPBGA256
- MPC5748C_MAPBGA324
- MPC5747C_LQFP176
- MPC5747C_MAPBGA256
- MPC5747C_MAPBGA324
- MPC5746C_LQFP176
- MPC5746C_MAPBGA256
- MPC5746C_MAPBGA324
- MPC5745C_LQFP176
- MPC5745C_MAPBGA256
- MPC5744C_LQFP176
- MPC5744C_MAPBGA256
- MPC5746B_LQFP176
- MPC5746B_MAPBGA256
- MPC5744B_LQFP176
- MPC5744B_MAPBGA256
- MPC5745B_LQFP176
- MPC5745B_MAPBGA256

- MPC5746C_MAPBGA100
- MPC5745C_MAPBGA100
- MPC5744C_MAPBGA100
- MPC5746B_MAPBGA100
- MPC5744B_MAPBGA100
- MPC5745B_MAPBGA100

2.4 Modules Configuration

Modules configurations were developed and tested using the Tresos Configuration Tool version "*EB tresos Studio 14.2.1 b140128-1223*"

Configuration definition files were developed according to AUTOSAR 4.0 Rev0003, 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=2, Power Architecture
- d=35, MPC574XG (derivative)
- m=10, Release major and minor version

- i=5, 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 1.0.5	Driver, Analog to Digital Conversion
Base	sw version 1.0.5	Base Module, General AUTOSAR and Hardware Specific register files
Can	sw version 1.0.5	Driver, Controller Area Network
CanIf	sw version 1.0.5	Support Stub, Controller Area Network Interface
Dem	sw version 1.0.5	Support Stub, Diagnostic Event Manager
Det	sw version 1.0.5	Support Stub, Development Error Tracer
Dio	sw version 1.0.5	Driver, Digital Input Output
EcuM	sw version 1.0.5	Support Stub, ECU State Manager
Eth	sw version 1.0.5	Driver, Ethernet
EthIf	sw version 1.0.5	Support Stub, Ethernet Interface
Fee	sw version 1.0.5	Driver, Flash EEPROM Emulation
Fls	sw version 1.0.5	Driver, Flash
Fr	sw version 1.0.5	Driver, FlexRay
FrIf	sw version 1.0.5	Support Stub, FlexRay Interface
Gpt	sw version 1.0.5	Driver, General Purpose Timer
Icu	sw version 1.0.5	Driver, Input Capture Unit
Lin	sw version 1.0.5	Driver, Local Interconnect Network
LinIf	sw version 1.0.5	Support Stub, Local Interconnect Network Interface
Mcl	sw version 1.0.5	Driver, Microcontroller Library (containing the DMA driver)
Mcu	sw version 1.0.5	Driver, Microcontroller Unit
MemIf	sw version 1.0.5	Support Stub, Memory Interface
Port	sw version 1.0.5	Driver, Port
Pwm	sw version 1.0.5	Driver, Pulse Width Modulation
Resource	sw version 1.0.5	Resource Module, Required by all other modules to select MCU derivative
Rte	sw version 1.0.5	Support Stub, only for Schedule Manager
Spi	sw version 1.0.5	Driver, Serial Peripheral Interface
Wdg	sw version 1.0.5	Driver, Watchdog
WdgIf	sw version 1.0.5	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_MCAL_<mod>_IM.pdf	Module Integration Manual
<mod>_TS_TtDdMmIiRr\doc\AUTOSAR_MCAL_<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)

Table continues on the next page...

Table 2-5. MCAL Sample Application Folder Structure (continued)

Folder or file	Description
- 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:

- Green Hills Multi 6.1.6 / Compiler 2014.1.6
- Windriver DIAB DIAB_5_9_4_8

2.8.1 GHS Compiler/Linker/Assembler Options

Table 2-6. Compiler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-ansi	Specifies ANSI C with extensions. This mode extends the ANSI X3.159-1989 standard with certain useful and compatible constructs.
-noSPE	Disables the use of SPE and vector floating point instructions by the compiler.
-Ospace	Optimize for size.
-sda=0	Enables the Small Data Area optimization with a threshold of 0.
-vle	Enables VLE code generation
-dual_debug	Enables the generation of DWARF, COFF, or BSD debugging information in the object file
-G	Generates source level debugging information and allows procedure call from debugger's command line.
--no_exceptions	Disables support for exception handling
-Wundef	Generates warnings for undefined symbols in preprocessor expressions
-Wimplicit-int	Issues a warning if the return type of a function is not declared before it is called
-Wshadow	Issues a warning if the declaration of a local variable shadows the declaration of a variable of the same name declared at the global scope, or at an outer scope
-Wtrigraphs	Issues a warning for any use of trigraphs
--prototype_errors	Generates errors when functions referenced or called have no prototype

Table continues on the next page...

Table 2-6. Compiler Options (continued)

Option	Description
--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
-c	Produces an object file (called input-file.o) for each source file.
--diag_error 223	Sets the specified compiler diagnostic messages to the level of error
--short_enum	Store enumerations in the smallest possible type
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DGHS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the GHS preprocessor symbol.
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.

Table 2-7. Assembler Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-G	Generates source level debugging information and allows procedure call from debugger's command line.
-list	Creates a listing by using the name of the object file with the .lst extension

Table 2-8. Linker Options

Option	Description
-cpu=ppc5748gz4204	Selects target processor: ppc5748gz4204
-cpu=ppc5748gz210	Selects target processor: ppc5748gz210
-nostartfiles	Do not use Start files.
-vle	Enables VLE code generation
--nocpp	Do not Generate Constructors/Destructors
-Mn	sort numerically the MAP file
-delete	The -delete option instructs the linker to remove functions that are not referenced in the final executable.
-ignore_debug_references	Ignores relocations from DWARF debug sections when using -delete. DWARF debug information will contain references to deleted functions that may break some third-party debuggers.
-keepmap	keeps the MAP file in case of link error

2.8.2 DIAB Compiler/Linker/Assembler Options

Table 2-9. Compiler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3V, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-Xdialect-ansi	Follow the ANSI C standard with some additions
-XO	Enables extra optimizations to produce highly optimized code
-g3	Generate symbolic debugger information and do all optimizations.
-Xsize-opt	Optimize for size rather than speed when there is a choice
-Xsmall-data=0	Set Size Limit for 'small data' Variables to zero.
-Xsmall-const=0	Set Size Limit for 'small const' Variables to zero.
-Xaddr-sconst=0x11	Specify addressing for constant static and global variables with size less than or equal to -Xsmall-const to far-absolute.
-Xaddr-sdata=0x11	Specify addressing for non-constant static and global variables with size less than or equal to -Xsmall-data in size to far-absolute.
-Xno-common	Disable use of the 'COMMON' feature so that the compiler or assembler will allocate each uninitialized public variable in the .bss section for the module defining it, and the linker will require exactly one definition of each public variable
-Xnested-interrupts	Allow nested interrupts
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xdebug-local-all	Force generation of type information for all local variables
-Xdebug-local-cie	Create common information entry per module
-Xdebug-struct-all	Force generation of type information for all typedefs, struct, union and class types
-Xforce-declarations	Generates warnings if a function is used without a previous declaration
-ee1481	Generate an error when the function was used before it has been declared
-Xmacro-undefined-warn	Generates a warning when an undefined macro name occurs in a #if preprocessor directive
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files
-W:as:,-l	Pass the option '-l' (lower case letter L) to the assembler to get an assembler listing file
-Wa,-Xisa-vle	Instruct the assembler to expect and assemble VLE (Variable Length Encoding) instructions rather than BookE instructions.
-c	Produces an object file (called input-file.o) for each source file.
-DAUTOSAR_OS_NOT_USED	-D defines a preprocessor symbol and optionally can set it to a value. AUTOSAR_OS_NOT_USED: By default in the package, the drivers are compiled to be used without Autosar OS. If the drivers are used with Autosar OS, the compiler option '-DAUTOSAR_OS_NOT_USED' must be removed from project options
-DDIAB	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the DIAB preprocessor symbol.

Table continues on the next page...

Table 2-9. Compiler Options (continued)

Option	Description
-DEU_DISABLE_ANSILIB_CALLS	-D defines a preprocessor symbol and optionally can set it to a value. This one defines the EU_DISABLE_ANSILIB_CALLS preprocessor symbol.

Table 2-10. Assembler Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3V, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-g	Dump the symbols in the global symbol table in each archive file.
-Xisa-vle	Expect and assemble VLE (Variable Length Encoding) instructions rather than Book E instructions. The default code section is named .text_vle instead of .text, and the default code section fill "character" is set to 0x44444444 instead of 0. The .text_vle code section will have ELF section header flags marking it as VLE code, not Book E code.
-Xasm-debug-on	Generate debug line and file information
-Xdebug-dwarf2	Generate symbolic debug information in dwarf2 format
-Xsemi-is-newline	Treat the semicolon (;) as a statement separator instead of a comment character.

Table 2-11. Linker Options

Option	Description
-tPPCE200Z4204N3VEN:simple	Sets target processor to PPCE200Z4204N3V, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-tPPCE200Z210N3VEN:simple	Sets target processor to PPCE200Z210N3, generates ELF using EABI conventions, No floating point support (minimizes the required runtime), selects simple environment settings for Startup Module and Libraries.
-Xelf	Generates ELF object format for output file
-m6	Generates a detailed link map and cross reference table
-Xlink-time-lint	Enable the checking of object and function declarations across compilation units, as well as the consistency of compiler options used to compile source files

Chapter 3
Known Issues for MPC574XG AUTOSAR 4.0 MCAL RTM 1.0.5

3.1 Known Issues

ID	Headline
LSMCAL-17130 CCB,n/a, Bug	[CAN] Inconsistant define for Can_FlexCan_SetUserAccessAllowed function prototype
LSMCAL-16895	[WDG] The Gpt timer may be stop wrongly

Chapter 4
Changes List for MPC574XG AUTOSAR 4.0 MCAL

4.1 RTM 1.0.5

ID	Subtype	Headline and Description
LSMCAL-11179	New Feature	[WDG] Analyze and implement the new features for servicing from fixed locations NewWork Description: On some platfroms the following bits were added to SWT_CR[SMD]: 10 Fixed Address Execution, the watchdog is serviced by executing code at the address loaded into the designated IAC register, which cannot be updated while the watchdog is enabled. 11 Incremental Address Execution, the watchdog is serviced by executing code at the address loaded into the designated IAC register, which can be updated. This seem to match to the AUTOSAR parameter WdgTriggerLocation which is currently Rejected in WDG requirements. Analyze this new feature, mark requirement as Approved and Fulfilled In, update xdm decription of paramter and implement this feature.
LSMCAL-12611	Bug	[Wdg] Compiler warning concerning division by zero is issued Problem detailed description (how to reproduce it): In configuration struct of SWT_OFF_MODE: CONST(Wdg_Swt_ConfigType, WDG_CONST) Wdg_Swt_OffModeSettings_Instance[!"substring-after($WdgSwtInstance,'SWT')"!]= { (uint32)(SWT_MAP_ENABLE_8BITS_U32 | SWT_WDG_DISABLED_U32 | SWT_WDG_SOFT_LOCK_U32), (uint32)0x00000100, /* Timeout config */ (uint32)0x00000000, /* Window config */ (uint32)0x00000000, /* Initial key value */ (uint32)0x00000000 /* Internal clock frequency */

Table continues on the next page...

ID	Subtype	Headline and Description
		}; Internal clock frequency = 0. In other source file we use internal clock frequency as detominator. It may pop a fail at building with message like: division by zero or even hang process. In current condition compiler optimize the division, therefore defect is only potential. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: Init or setmode to offmode Observed behavior: It may have a compiler warning, fail at building or freeze Expected behavior: Defect should be fixed, no problem when using offmode Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Fix default value off internal frequency
LSMCAL-14920	Bug	[PWM] OPWMT interrupt should be disabled if notifications are not used Problem detailed description (how to reproduce it): For EMIOS channels confiugred in OPWMT mode, the interrupts are always enabled(FEN bit is set) and routed to DMA (DMA bit is set) . If notifications are not configured for the channel, there is no interrupt code to clear the FLAG bit after a trigger event occurs. This causes the EMIOS to send a flood of triggers. Preconditions: EMIOS channel in OPWMT mode, notifications disabled. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Flood of triggers setn by EMIOS (for example, to BCTU) Expected behavior: No flooding of triggers should occur. The trigger signals should happen only once per PWM period. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Disable the interrupts for OPWMT channels in case notifications are not used. dev_engr387973_nxf33902_pwm_OPWMT_notifications_are_not_used
LSMCAL-14931	Bug	[MCU] Some functions from FIRC and SIRC are not used on all platforms Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Some functions which was declared in below layer, but was not used in upper layer. For example : Mcu_FIRC_ActiveClockStatus + Mcu_FIRC_StandbyClockStatus in

Table continues on the next page...

ID	Subtype	Headline and Description
		Mcu_FIRC.c Mcu_SIRC_ActiveClockStatus + Mcu_SIRC_StandbyClockStatus in Mcu_SIRC.c As I know if a function was declared in below layer but not used in upper layer, it should be removed or round up. 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): Should validate if it affect to specific. Should add CR to BKICR of all config affected
LSMCAL-14982	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)
LSMCAL-15331	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.

Table continues on the next page...

ID	Subtype	Headline and Description
		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 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 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): [...]
LSMCAL-15490	Bug	[WDG] Wrong reference to non-autosar parameter '-- Converted from JIRA ticket MCAL-15101 -- These ConfigAffected: S32GXX_4.2 were removed as they are invalid in CQ currently. 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 DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/TS_T2D47M10I0R0/Wdg/WdgClockReferencePoint 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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): Current solution: name="REF" value="ASPathDataOfSchema:/AUTOSAR/EcucDefs/Wdg/WdgClockReferencePoint" / Changed to: name="REF" value="ASPathDataOfSchema:/TS_T2D47M10I0R0/Wdg/WdgClockReferencePoint" /
LSMCAL-15588	Bug	[WDG] Wrong defines for AR RELEASE version check in Wdg - clone of LSMCAL-14854 [ENGR00387742] 'Converted from JIRA ticket -MCAL-15095 The Reported Baseline 'DUMMY_VERSION' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. 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 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
LSMCAL-15629	Bug	[WDG] Missing #endif in the function Wdg_ChannelValidateMode '-- Converted from JIRA ticket MCAL-15638 -- Detailed description (how to reproduce it): This is an internal defect, it does not affect any customer delivery. In the function Wdg_ChannelValidateMode(), statements below was missing #endif { #if (WDG_DISABLE_ALLOWED == STD_OFF)

Table continues on the next page...

ID	Subtype	Headline and Description
		if((boolean)TRUE == disableRejectedErrActive) { Dem_ReportErrorStatus((Dem_EventIdType)Wdg_E_Disable_Rejected.id, DEM_EVENT_STATUS_FAILED); } } Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Wdg driver build fail Expected behavior: Wdg driver is not build fail Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add #endif for ASR 4.0.3
LSMCAL-15743	Bug	[MCU] CMU should be enabled after the clock input (source) is stable Converted from JIRA ticket -MCAL-16761 Detailed description (how to reproduce it): In Mcu_InitClock function, it calls Mcu_IPW_InitClock function. After the function Mcu_IPW_InitClock configure the clock tree, if the system clock which was configured at the configuration time is IRC or XOSC, then it enables CMU without checking the clock source status of CMUs. Because of this, the CMU may report the source clock is out of range. The driver should check the corresponding clock input status (PLLs status, XOSC status, etc) before enabling CMUs. For more details about the clock input, please refer the "Table 5-8. Clock input sources" in RM Observed behavior: Spurious interrupt occurs due to the FLLI flag is set immediately after CMUs are enabled Expected behavior: No Spurious interrupt, the FLLI flag is not set immediately after CMUs are enabled Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):
LSMCAL-15888	Bug	[WDG]Fixing wrong macro for defining mode configuration Converted from JIRA ticket -MCAL-16980 The Reported Baseline 'DUMMY_VERSION' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. No valid config left, ALL/GENERAL used instead. Detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		Build failed due to initilize mode configuration 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
LSMCAL-16064	New Feature	[MCU] Disable specific core on Standby exit Initial Description: On standby exit, the core (z2, z4a, z4b) will be reset and run depend on the DRUN bit in corresponding CCTLx registers. In the customer ECU, after standby exit, the core is configured to started from BAF entry. And the BAF allows only one core running so the customer needs to clear DRUN bit of other CCTLx before entering Standby mode. Can Mcu driver support these configuration? The idea is to add one parameter for each core into McuCoreConfiguration container in McuModeSettingConf, if the parameter is enabled then the Mcu_SetMode will clear DRUN bit of corresponding CCTL register right before entering Standby mode. e.g: "Core Z4a DRUN Clear on Standby exit"/"Core Z4b DRUN Clear on Standby exit"/"Core Z2 DRUN Clear on Standby exit", if these parameters are enabled and checked, Mcu_SetMode will clear DRUN bit in CCTL1, CCTL2, and CCTL3 respectively when entering Standby mode. (Note: After the CR type is decided, will remain only one section from the two listed below, the other one will be deleted)
LSMCAL-16104	New Feature	[MCU] Investigate cyclomatic complexity and nesting level NewWork Description: Requirements: - 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. 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

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: Quality (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): In AUTOSAR_MCAL_MDL_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments - for all the cyclomatic complexity less than 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity greater than 20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth greater than 4 the problems shall be resolved
LSMCAL-16324	New Feature	[IPV_MCV4] Disable specific core on Standby exit Initial Description: On standby exit, the core (z2, z4a, z4b) will be reset and run depend on the DRUN bit in corresponding CCTLx registers. In the customer ECU, after standby exit, the core is configured to started from BAF entry. And the BAF allows only one core running so the customer needs to clear DRUN bit of other CCTLx before entering Standby mode. Can Mcu driver support these configuration? The idea is to add one parameter for each core into McuCoreConfiguration container in McuModeSettingConf, if the parameter is enabled then the Mcu_SetMode will clear DRUN bit of corresponding CCTL register right before entering Standby mode. e.g: "Core Z4a DRUN Clear on Standby exit"/"Core Z4b DRUN Clear on Standby exit"/"Core Z2 DRUN Clear on Standby exit", if these parameters are enabled and checked, Mcu_SetMode will clear DRUN bit in CCTL1, CCTL2, and CCTL3 respectively when entering Standby mode. (Note: After the CR type is decided, will remain only one section from the two listed below, the other one will be deleted)
LSMCAL-16443	Bug	[WDG] The check of WdgTimeoutPeriod is incorrect Problem detailed description (how to reproduce it): During the configuration, the driver checks whether WdgTimeoutPeriod is below 0.002 second and throws error message. expr="(node:fallback(., 0.01) less 0.002) = 'true' " true="WdgTimeoutPeriod must be higher than 0.002"/> This is not correct for MPC574XR device where the SWT clock is 16Mhz FIRC. With minimum valule of SWT_TO = 0x100, the minimum period could be 16us

Table continues on the next page...

ID	Subtype	Headline and Description
		= (0.000016s)
LSMCAL-16535	Bug	[ENET][ETH] the same buffer is allocated twice before it is released Converted from JIRA ticket -MCAL-18150 The Reported Baseline 'BLN_SMCAL_4.2_S32K14X_RTM_1.0.0' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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): These code lines will be moved to the location after setting Bit R(ready) in TxBD /* Write frame/buffer status bits */ if((VAR(boolean, AUTOMATIC))TRUE == bConfirm) \{ /* TxConfirmation is requested, also set TxConfirmation bit */ Eth_au8TxBufFlags[u8CtrlIdx][u8Buf] = ENET_TXB_LOCK_U8 | ENET_TXB_FIRST_U8 | ENET_TXB_LINK_U8 | ENET_TXB_CONF_U8 |

Table continues on the next page...

ID	Subtype	Headline and Description
		ENET_TXB_INTRANSMIT_U8; /* Update number of pending confirmations */ /* Enter critical section for RMW access */ SchM_Enter_Eth_ETH_EXCLUSIVE_AREA_00(); Eth_u8LockedTxBufCount[u8CtrlIdx]++; /* Exit Exclusive area */ SchM_Exit_Eth_ETH_EXCLUSIVE_AREA_00(); } else \{ Eth_au8TxBufFlags[u8CtrlIdx][u8Buf] = ENET_TXB_LOCK_U8 | ENET_TXB_FIRST_U8 | ENET_TXB_LINK_U8; }
LSMCAL-16570	Bug	[GPT] Implement the requirement PR-MCAL-3307 (Gpt_ChangeNextTimeoutValue API) '- Converted from JIRA ticket MCAL-18194 -- The Reported Baseline 'BLN_SMCAL_4.2_S32K14X_RTM_1.0.0' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): The API Gpt_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)
LSMCAL-16595	New Feature	[LIN] Fix naming variable rules follow SMCAL_Coding NewWork Description: Following rule 2.22 in SMCAL_Coding. The global variables shall use following naming convention: <Msn>_[<Ip>_] [<PrefixType>]<VarName> Requirement source: In file Lin_IPW.c and Lin_IPW.h has a variable pcPduInfoPtr which is incorrect with 2.22 rule. Proposed solution (Optional): Change variable pcPduInfoPtr to pPduInfoPtr in file Lin_IPW.c and Lin_IPW.h for a lot modules.
LSMCAL-16617	Bug	[ADC] Missing sign file for Adc_VersionCheck_Inc.m in Adc.mak generate_PC/Adc_VersionCheck_Inc.m this file is existing in the plugin but it did not list in the list of signed files This ticket is raised for next release, follow up to fix. But it not impact to customer package due to this issue already workaround by integrator.

Table continues on the next page...

ID	Subtype	Headline and Description
LSMCAL-16718	Bug	[SPI] Missing element of struct type in Spi_aSpiJobTSBConfig Problem detailed description (how to reproduce it): In Spi_aSpiJobTSBConfig of Spi_RegOperations.m The path below is incorrect: ../../../SpiNonAUTOSAR/SpiITSBModeSupport = 'true' The element(bIsITSBmode) doesn't generated when SpiITSBModeSupport = true. Preconditions: - In config file(Spi.xdm): Exist SpiITSBModeSupport Test Case ID (internal TC that caught the defect) - optional - Spi_TS_019 Trigger: NA Observed behavior: The element(bIsITSBmode) will not generated when SpiITSBModeSupport = true. File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 728, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 756, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 784, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 812, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 840, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). File"e:\local_02\output\projects\ar_misra_spi_ghs\target\build \Spi_TS_019_cfgPB_CORE2\generate\src\Spi_PBcfg.c", line 868, Violates [MISRA 2004 Rule 9.2, required]: no MISRA violation comment was found (maybe wrong format is used). Expected behavior: The element(bIsITSBmode) generated when SpiITSBModeSupport = true. No error misra. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): 1. Replace: [!IF "../../../SpiNonAUTOSAR/SpiITSBModeSupport = 'true' and ecu:get('SpiITSBSupport')='TRUE'"!][!// with [!IF "../../../../SpiNonAUTOSAR/SpiITSBModeSupport = 'true' and

Table continues on the next page...

ID	Subtype	Headline and Description
		ecu:get('SpiITSBSupport')='TRUE'"!][!//
LSMCAL-16721	Bug	[LIN][IPV_LINFLEX] Misra error on Matterhorn RTM 1.0.2 Problem detailed description (how to reproduce it): Misra log has an error like as below: "File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223_reg_02\plugins \Lin_TS_T2D36M10I2R0\src\Lin_LINFlex.c", line 1185, Violates [MISRA 2004 Rule 1.2, required]: no relevant comment was found for MISRA violation on this line. File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223_reg_02\plugins \Lin_TS_T2D36M10I2R0\src\Lin_LINFlex.c", line 1185, Violates [MISRA 2004 Rule 12.2, required]: no relevant comment was found for MISRA violation on this line." For more details about MISRA comments see "MisraView_Guide.pdf" document and file lin_misra.zip. Preconditions: Run misra tool Test Case ID (internal TC that caught the defect) - optional Any test case Trigger: NA Observed behavior: Rule 1.2, 12.2 Expected behavior: To fix these errors 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.
LSMCAL-16730	Bug	[MCL] Wrong configure Crossbar when PRSx registers are ReadOnly Problem detailed description (how to reproduce it): If PRSx registers are ReadOnly and Crossbar is missing some Masters (has master ID not continuous) then cann't configure Crossbar feature in EbTresos. Example for Matterhorn: -the PRS is readonly: Mcl.Crossbar.PRS.IsReadOnly:1 -Missing some Masters as: Mcl.Crossbar.AXBS0.Masters:Master1,Master2,Master3,Master4,Master5 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Cannot Configue the Crossbar feature Expected behavior: User can configure the Crossbar feature

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 Please see attachment for more detail.
LSMCAL-16804	Bug	"[WDG] WdgClkSrcRef uses autosar instead of vendor specific short-name path -- Converted from JIRA ticket MCAL-18609 -- These ConfigAffected: LEGACY_PLATFORM were removed as they are invalid in CQ currently. No valid config left, ALL/GENERAL used instead. Detailed description (how to reproduce it): Issue with our epd file: Wdg.epd In current code (RTM 2.0.0, Fee.epd) the reference is defined as follows: <ECUC-REFERENCE-DEF UUID="ECUC:f111edc2-382d-4d8e-9e96-7b23c31c31db"> <SHORT-NAME>WdgClkSrcRef</SHORT-NAME> ... <DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/ AUTOSAR/EcucDefs/Wdg/WdgClockReferencePoint</DESTINATION-REF> ... </ECUC-REFERENCE-DEF> But customer requests it to be as follows (since it points to vendor specific container which has no counterpart in StMD): <ECUC-REFERENCE-DEF UUID="ECUC:f111edc2-382d-4d8e-9e96-7b23c31c31db"> <SHORT-NAME>WdgClkSrcRef</SHORT-NAME> ... <DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/ TS_T2D30M20I0R0/Wdg/WdgClockReferencePoint</DESTINATION-REF> ... </ECUC-REFERENCE-DEF> In AUTOSAR_TPS_ECUConfiguration.pdf please see the section 5.1 "Deriving vendor specific module definitions from standardized module definitions", Example 5.3 (<DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/VendorY/CanDrv/CanDrvTrcvChannel</DESTINATION-REF>). And also see this requirement which is violated: [ecuc_sws_6046] A pure vendor specific reference definition (which has no counterpart in the STMD) can refer either to a standardized container (has a counterpart in the STMD) or to a vendor specific container. If the reference points to a standardized container the standardized AUTOSAR short-name path shall be used. If the reference points to the vendor specific container the fully qualified vendor specific short-name path shall be used.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: Wdg.epd does not follow ecuc_sws_6046 in AUTOSAR_TPS_ECUConfiguration.pdf Expected behavior: [ecuc_sws_6046 is followed] Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [ecuc_sws_6046] require to have the following simple modification in Wdg.xdm for WdgClkSrcRef Instead of: <a:da name="REF" value="ASPathDataOfSchema:/AUTOSAR/EcucDefs/Wdg/WdgClkSrcRef"/> We should use <a:da name="REF" value="ASPathDataOfSchema:/M4_XDM_AR_PKG_NAME/Wdg/WdgClkSrcRef"/>"
LSMCAL-16970	Bug	[IPV_ADCDIG] Missing the mask of injected channel conversion and the mask of conversion for the CTU channel Problem detailed description (how to reproduce it): When run the tests of ADC, the problem will be occur. Because ADCDIG_ISR_END_CHANNEL_INJ_CONV_MASK_U32 and ADCDIG_ISR_END_CHANNEL_CTU_CONV_MASK_U32 are not define in Adc_Reg_eSys_Adcdig.h Expected behavior: The test of ADC should be run completely Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add /** * @brief JEOC. * @details End of injected channel conversion. */ #define ADCDIG_ISR_END_CHANNEL_INJ_CONV_MASK_U32 (0x00000008UL) /** * @brief EOCTU. * @details End of CTU conversion. * It is the interrupt of the digital end of conversion for the CTU channel; * active when set. */ #define ADCDIG_ISR_END_CHANNEL_CTU_CONV_MASK_U32 (0x00000010UL)"

Table continues on the next page...

ID	Subtype	Headline and Description
LSMCAL-16975	Bug	[IPV_STM][GPT] Implement the requirement PR-MCAL-3307 (Gpt_ChangeNextTimeoutValue API) -- Converted from JIRA ticket MCAL-18194 -- The Reported Baseline 'BLN_SMCAL_4.2_S32K14X_RTM_1.0.0' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): The API Gpt_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)
LSMCAL-16976	Bug	[IPV_CMU][MCU] CMU should be enabled after the clock input (source) is stable Converted from JIRA ticket -MCAL-16761 Detailed description (how to reproduce it): In Mcu_InitClock function, it calls Mcu_IPW_InitClock function. After the function Mcu_IPW_InitClock configure the clock tree, if the system clock which was configured at the configuration time is IRC or XOSC, then it enables CMU without checking the clock source status of CMUs. Because of this, the CMU may report the source clock is out of range. The driver should check the corresponding clock input status (PLLs status, XOSC status, etc) before enabling CMUs. For more details about the clock input, please refer the "Table 5-8. Clock input sources" in RM Observed behavior: Spurious interrupt occurs due to the FLLI flag is set immediately after CMUs are enabled Expected behavior: No Spurious interrupt, the FLLI flag is not set immediately after CMUs are enabled Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
LSMCAL-16977	Bug	[IPV_MCV4][MCU] CMU should be enabled after the clock input (source) is stable Converted from JIRA ticket -MCAL-16761 Detailed description (how to reproduce it): In Mcu_InitClock function, it calls Mcu_IPW_InitClock function. After the function Mcu_IPW_InitClock configure the clock tree, if the system clock which was configured at the configuration time is IRC or XOSC, then it

Table continues on the next page...

ID	Subtype	Headline and Description
		enables CMU without checking the clock source status of CMUs. Because of this, the CMU may report the source clock is out of range. The driver should check the corresponding clock input status (PLLs status, XOSC status, etc) before enabling CMUs. For more details about the clock input, please refer the "Table 5-8. Clock input sources" in RM Observed behavior: Spurious interrupt occurs due to the FLLI flag is set immediately after CMUs are enabled Expected behavior: No Spurious interrupt, the FLLI flag is not set immediately after CMUs are enabled Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
LSMCAL-16985	Bug	[IPV_DMA][MCL] Inconsistent usage of TCD address type -- Converted from JIRA ticket MCAL-17263 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): In the MCL APIs the TCD address is sometimes used as uint32, othertimes as pointer to uint32, the APIs should be updated to have a consistent approach. Example: Mcl_DmaGetChannelTcdAddress returns uint32 Mcl_DmaConfigTcd uses pointer to uint32 parameter Preconditions: usage of TCD APIs Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: In the MCL APIs the TCD address is sometimes used as uint32, othertimes as pointer to uint32. Expected behavior: MCL APIs should have a consistent approach of TCD address type. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): MCL APIs should have a consistent approach of TCD address type.
LSMCAL-16995	Bug	[ICU] Fix wrong clear bit flag interrupt -- Converted from JIRA ticket MCAL-18686 -- The Reported Baseline 'BLN_SMCAL_4.0_S32K14X_RTM_1.0.2' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		Some IPVs write bit 1 to bit position flag interrupt from Register Flag Interrupt. If using functions REG_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
LSMCAL-16998	Bug	[FLS] There is an inconsistency of the xdm class attribute for the flash sector parameters -- Converted from JIRA ticket MCAL-18678 -- The Reported Baseline 'BLN_SMCAL_4.0_S32K14X_RTM_1.0.2' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. 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 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.

Table continues on the next page...

ID	Subtype	Headline and Description
LSMCAL-17017	New Feature	[IPV_STM] [GPT] Update to violate misra Rule 9.2 Problem detailed description (how to reproduce it): Line of code, refer to driver: static VAR(uint32, GPT_VAR) Gpt_Stm_u32NextTargetValue[GPT_STM_MODULES_NUM_U8] [GPT_STM_MODULE_CHAN_NUM_U8] = {{(uint32)0,(uint32)0}}; Misra violation: Rule 9.2(advisory): Violates MISRA 2004 Required Rule 9.2, Braces shall be used to indicate and match the structure in the * non-zero initialisation of arrays and structures. At here, we tend to initialized for whole structure, however, * when configure only 1 controller is configured, tool reported as non-zero initialisation. N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: Misra report generation Observed behavior: Without Misra report generated. Expected behavior: Misra report is generate sucessful. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Just add comment before line of code to violate misra 9.2. With misra log, please see more at attachment field.
LSMCAL-17023	New Feature	[MCU] Update the default value of McuClkSetSXOSC_En and warnings Some customer got issue with SXOSC in MCU configuration. The default value of SXOSC parameter McuClkSetSXOSC_En is true, so usually the customer will leave this parameter checked and also enable McuSXOscControl in McuModeSettingConf container. But the customer board does not have slow crystal so the MCU clock setting and mode transition could not be done. The customer expects that MCU will throw some warnings or messages to the customer if the SXOSC is enabled in clock configuration (McuClkSetSXOSC_En is true) or in Mode configuration (McuSXOscControl) but the clock is not used or referenced in MCU driver. e.g: 1. Change the default value of McuClkSetSXOSC_En and McuSXOscControl to false 2. Add the validation for McuSXOscControl and McuClkSetSXOSC_En, if these parameters are enabled, but the SXOSC clock is used in clock reference point, or source of clock out then the MCU plugin will throw warnings to the customer like: SXOSC is enabled but is not used as a McuClockReferencePoint or source of clock out McuAuxClock6. Please check whether SXOSC is available on the board. 3. Add some more informaiton to the description of these parameters (McuClkSetSXOSC_En and McuSXOscControl) to inform the customers that

Table continues on the next page...

ID	Subtype	Headline and Description
		these parameters can be enabled only when the slow crystal is designed on the board. In Gpt driver, the driver may check SXOSC if this is used as source for RTC too.
LSMCAL-17026	New Feature	[IPV_PIT] [GPT] Update to violate misra Rule 12.4 Problem detailed description (how to reproduce it): Line of code, refer to driver: static VAR(uint32, GPT_VAR) Gpt_Stm_u32NextTargetValue[GPT_STM_MODULES_NUM_U8] [GPT_STM_MODULE_CHAN_NUM_U8] = {{(uint32)0,(uint32)0}}; Misra violation: Rule 9.2(advisory): Violates MISRA 2004 Required Rule 9.2, Braces shall be used to indicate and match the structure in the * non-zero initialisation of arrays and structures. At here, we tend to initialized for whole structure, however, * when configure only 1 controller is configured, tool reported as non-zero initialisation. N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: Misra report generation Observed behavior: Without Misra report generated. Expected behavior: Misra report is generate sucessful. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Just add comment before line of code to violate misra 9.2. With misra log, please see more at attachment field.
LSMCAL-17027	New Feature	[LIN][IPV_LINFLEX] LINSR shall be cleared on a spurious interrupt According to the PR----MCAL-3272---.lin 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. However, in "Lin_LINFlex.c" we have: {code:none} /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer * * @violates @ref Lin_LINFlex_c_REF_9 cast should not be performed */ u32Lin_status = REG_READ32(LINFLEX_LINIER_ADDR32(u8Lin_LINFlex)); /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer * * @violates [~nxf32113]ef Lin_LINFlex_c_REF_9 cast should not be

Table continues on the next page...

ID	Subtype	Headline and Description
		performed */ u32Lin_status &= REG_READ32(LINFLEX_LINSR_ADDR32(u8Lin_LINFlex)); if (0UL != u32Lin_status) { /* Some IP-specific logic in here */ } else { /* do nothing */ } {code} Proposed Solution: * On the "else" branch, LINSR should be cleared. For more information: * PR-MCAL-3272.lin: 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. * Verify: test case
LSMCAL-17037	Bug	[SPI][IPV_DSPI] Misra error Detailed description (how to reproduce it): In Spi_DSPI.c file has an misra error (Rule 11.4, advisory). Please see the attachment to get more detail information Preconditions: Generate any tests and run misra tool (code_analysis) Test Case ID (internal TC that caught the defect) - optional Any tests Observed behavior: Misra error 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): Fill comment for these lines which has errors.
LSMCAL-17039	Bug	[MCU] Compiler Warning in Calypso RTM 1.0.5 Detailed description (how to reproduce it): Mcu driver has some compiler warning. Please see the attachment to get more detail information. Preconditions: Generate any tests and run CW tool (cwe) Test Case ID (internal TC that caught the defect) - optional Any tests Observed behavior: Compiler warning Expected behavior: No warning

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): Please see the analysis tab
LSMCAL-17074	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):
LSMCAL-17077	New Feature	[GPT] Update the default value of McuClkSetSXOSC_En and warnings Some customer got issue with SXOSC in MCU configuration. The default value of SXOSC parameter McuClkSetSXOSC_En is true, so usually the customer will leave this parameter checked and also enable McuSXOscControl in McuModeSettingConf container. But the customer board does not have slow crystal so the MCU clock setting and mode transition could

Table continues on the next page...

ID	Subtype	Headline and Description
		not be done. The customer expects that MCU will throw some warnings or messages to the customer if the SXOSC is enabled in clock configuration (McuClkSetSXOSC_En is true) or in Mode configuration (McuSXOscControl) but the clock is not used or referenced in MCU driver. e.g: 1. Change the default value of McuClkSetSXOSC_En and McuSXOscControl to false 2. Add the validation for McuSXOscControl and McuClkSetSXOSC_En, if these parameters are enabled, but the SXOSC clock is used in clock reference point, or source of clock out then the MCU plugin will throw warnings to the customer like: SXOSC is enabled but is not used as a McuClockReferencePoint or source of clock out McuAuxClock6. Please check whether SXOSC is available on the board. 3. Add some more informaiton to the description of these parameters (McuClkSetSXOSC_En and McuSXOscControl) to inform the customers that these parameters can be enabled only when the slow crystal is designed on the board. In Gpt driver, the driver may check SXOSC if this is used as source for RTC too.
LSMCAL-17100	Bug	[ADC] The DMA transmission is stopped when the DMA ISR is preempted by higher priority interrupts Problem detailed description (how to reproduce it): The customer configured ADC group in continuous mode with DMA transfer One DMA interrupt is blocked by other ISR or debugger, the next 2 DMA interrupts will be close to each other(as shown below). In the 1st DMA interrupt, when DMA channel is enabled again, the last blocked ADC->DMA request will be executed immediately. After the DMA transmission. the 2nd DMA interrupt will be generated, which is very close to the end of the 1st DMA interrupt handler. In the 2nd DMA interrupt, ADC_ISR will be checked again, however, it is possible that the flag may have been cleared by the 1st DMA interrupt. In this case, DMA interrupt will do nothing and exit handler directly. So DMA channel will keep disabled and DMA continuous transmission will stop. Observed behavior: The DMA channel for ADC transfer is stopped incorrectly in debug mode or interrupted by higher priority interrupts. Expected behavior: The DMA channel for ADC transfer should be enabled even in debug mode or interrupted by higher priority interrupts. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The flag ADCDIG_ISR_END_CHAIN_NORM_CONV_U32 is checked but Adc is continuous conversion mode so the flag would be asynchronous set and would be cleared while the DMA transmission is still disabled. if ((ADCDIG_ISR_END_CHAIN_NORM_CONV_U32 == (u32IsrValue & ADCDIG_ISR_END_CHAIN_NORM_CONV_U32)) && (SwNormalQueueIndex > 0U)) \{ Group = Adc_aUnitStatus[Unit].SwNormalQueue[0]; /* Enter critical region */ SchM_Enter_Adc_ADC_EXCLUSIVE_AREA_13(); /** @violates @ref Adc_Adcdig_Irq_c_REF_9 cast from unsigned int to pointer */ /** @violates

Table continues on the next page...

ID	Subtype	Headline and Description
		@ref Adc_Adcdig_Irq_c_REF_3 A cast should not be performed between a pointer type and an integral type. */ REG_WRITE32(ADCDIG_ISR_REG_ADDR32(Unit), ADCDIG_ISR_END_CHANNEL_NORM_CONV_CLEAN_U32| ADCDIG_ISR_END_CHAIN_NORM_CONV_CLEAN_U32); Adc_Adcdig_DmaEndNormalConv(Unit, Group); }
LSMCAL-17138	New Feature	[ADC] Missing comment in adc.xdm Detailed description (how to reproduce it): Missing some comments so the TraceabilityMatrix report have some warnings Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Observed behavior: N/A Expected behavior: The TraceabilityMatrix report have not warning Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add "@implement" comment into adc.xdm

4.2 RTM 1.0.4

ID	Subtype	Headline and Description
ENGR00388680	Defect	[ADC] ADC Mux Delay is casted to 8 bit data 'Detailed description (how to reproduce it): Below is the report from the customer. Doing some tests, we changed the values of the ADC_0_DSDR register and we noticed that something was not properly configured. -We modified the Tresos parameter value "ADC Mux Delay" to 1023. -As per reference manual chapter 34.5.31, this is a 16-bit register. -But when we check the register in Trace32 we saw that was set to 255. -Checking the auto generated code we have seen that the value configured in TRESOS is properly generated but is cast to uint8 (the reason to see 0x00FF instead of 0x03FF). /**< @brief Mux delay value */ (ADCDIG_MUX_DELAY_VALUE_U8((uint8)1023)), Preconditions: The ADC Mux Delay is configured with value greater than 0xFF. Test Case ID (internal TC that caught the defect) - optional N/A Observed behavior: The ADC Mux Delay is casted to 8 bit data. The expected data is not written to the corresponding register.

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: The expected data is not written to the corresponding register. 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: - Instead of ADCDIG_MUX_DELAY_VALUE_U8, the ADCDIG_MUX_DELAY_VALUE_U16 should be defined for an unsigned 16 bit value. -The Tresos macro should not cast the configured value of the parameter (AdcMuxDelay) to an unsigned 8 bit.(or it could be eliminated).",'ADCDIG_MUX_DELAY_VALUE_U16 should be defined for an unsigned 16 bit value. Both of in generation macro and driver register define.,ENGR00388747
ENGR00390138	Defect	[ADC] Adc configuration is not generated when BCTU feature is enabled 'Problem detailed description (how to reproduce it): The customer enabled BCTU feature with ADC and they got following errors at the generation time. SHORTMESSAGELOCALIZED [C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". SHORTMESSAGENONLOCALIZED [C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". LONGMESSAGELOCALIZED Parsing file "C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c", line "91" [C:\NXP\AUTOSAR\MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins \Adc_TS_T2D35M10I3R0/generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". LONGMESSAGENONLOCALIZED Parsing file "C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c", line "91" [C:\NXP\AUTOSAR\MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins \Adc_TS_T2D35M10I3R0/generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". SHORTMESSAGEENRICHEDLOCALIZED [C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". SHORTMESSAGEENRICHEDNONLOCALIZED [C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". LONGMESSAGEENRICHEDLOCALIZED Parsing file "C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/ generate_PC/src/Adc_Cfg.c", line "91" [C:\NXP\AUTOSAR\MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins \Adc_TS_T2D35M10I3R0/generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". LONGMESSAGEENRICHEDNONLOCALIZED Parsing file "C:\NXP\AUTOSAR \MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins\Adc_TS_T2D35M10I3R0/

Table continues on the next page...

ID	Subtype	Headline and Description
		generate_PC/src/Adc_Cfg.c", line "91" [C:\NXP\AUTOSAR\MPC574XG_MCAL4_0_RTM_1_0_3\eclipse\plugins \Adc_TS_T2D35M10I3R0/generate_PC/src/Adc_Cfg.c:91]: Failed to access include file "Adc_Bctu_Cfg.m". Preconditions: - BCTU feature is enabled and configured in ADC driver Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: - Failed to generate Adc configuration code. Expected behavior: - The configuration code is generated successfully without error message. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The Adc_Bctu_Cfg.m files in Adc_TS_T2D35M10I3R0/generate_PC and Adc_TS_T2D35M10I3R0/generate_PB were not signed for tresos license.",'missing signed Adc_Bctu_Cfg.m from PB and PC.
ENGR00389925	Defect	[ADC] Build failed when Enable MHT, all groups are without interrupt groups. All AdcInterrupt were not enabled. '-- Converted from JIRA ticket MCAL-15116 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Enable MHT, all groups are without interrupt groups. All AdcInterrupt were not enabled. #if ((defined(ADC_UNIT_0_ISR_USED)) || (defined(ADC_UNIT_1_ISR_USED)) || \ (defined(ADC_UNIT_2_ISR_USED)) || (defined(ADC_UNIT_3_ISR_USED)) || \ (defined(ADC_DMA_SUPPORTED)) \ ) #if (ADC_GRP_NOTIF_CAPABILITY == STD_ON) LOCAL_INLINE FUNC(boolean, ADC_CODE) Adc_Adcdig_CheckNotification (VAR(Adc_GroupType, AUTOMATIC) Group) >> This function was not built. But "C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/ Adc_Adcdig_Irq.c", line 2155, function Adc_Adcdig_EndMultipleCtuConv, program calls Adc_Adcdig_CheckNotification: bCallNotification = Adc_Adcdig_CheckNotification(Group); ==> Build failed. Test Case ID (internal TC that caught the bug) - optional Adc_TS_Eq_Cot_01",'N/A
ENGR00389927	Defect	[ADC] Build failed when EnableCtuTrigNonAutosarApi is disabled. SetChannel is enabled. HW trigger is disabled. BCTU control mode is enabled. '-- Converted from JIRA ticket MCAL-15115 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features):

Table continues on the next page...

ID	Subtype	Headline and Description
		AdcEnableCtuTrigNonAutosarApi is disabled. SetChannel is enabled. HW trigger is disabled. BCTU control mode is enabled. In BCTU.c: #if ((ADC_ENABLE_CTUTRIG_NONAUTO_API == STD_ON) || (ADC_HW_TRIGGER_API == STD_ON)) extern P2CONST(Adc_ConfigType , ADC_VAR, ADC_APPL_CONST) Adc_pCfgPtr; static VAR(uint16, AUTOMATIC) Adc_Bctu_u16ListAddressIndex = 0U; static VAR(uint16, AUTOMATIC) Adc_Bctu_au16ListChannelAddress[ADC_MAX_HW_UNITS] = {0U}; static VAR(uint16, AUTOMATIC) Adc_Bctu_aPreListElement[ADC_MAX_HW_UNITS] = {0U}; ==> Adc_pCfgPtr is not externed, Adc_Bctu_u16ListAddressIndex, Adc_Bctu_au16ListChannelAddress, Adc_Bctu_aPreListElement are not declared In BCTU.h: #if ((ADC_ENABLE_CTUTRIG_NONAUTO_API == STD_ON) || (ADC_HW_TRIGGER_API == STD_ON)) FUNC(void, ADC_CODE) Adc_BctuUpdateListElement ==> Adc_BctuUpdateListElement is not declared. ==> Build failed as below: STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Bctu.c", line 1195: error #20: STDERR: identifier "Adc_pCfgPtr" is undefined STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Bctu.c", line 1198: error #20: STDERR: identifier "Adc_Bctu_u16ListAddressIndex" is undefined STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Bctu.c", line 1200: error #20: STDERR: identifier "Adc_Bctu_au16ListChannelAddress" is undefined STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Bctu.c", line 1209: error #223-D: STDERR: function Adc_BctuUpdateListElement declared implicitly STDERR: identifier "Adc_Bctu_aPreListElement" is undefined STDERR: Adc_Bctu_aPreListElement[Adc_pCfgPtr->pGroups[Group].HwUnit] = ChannelCount; Test Case ID (internal TC that caught the bug) - optional Adc_TS_Eq_Cot_01",'N/A
ENGR00390841	Defect	[ADC] Double buffering does not work properly when enable cache for DMA 'Problem detailed description (how to reproduce it): When enable cache for DMA, Double buffering for ADC,DMA does not properly, It stops after passing some data, we need clear cache after call noityfication for DMA. Preconditions: Enable Cache, only effect to platform have cache Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: DMA work wrong Expected behavior: DMA work when enable double buffering and cache for DMA Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		We can work around this issue by Change from: #if (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING) /** @violates @ref Adc_Adcdig_Irq_c_REF_5 Array indexing shall be the only allowed form of pointer arithmetic */ if ((boolean) TRUE == Adc_pCfgPtr->pGroups[Group].bAdcDoubleBuffering) { DmaChannel = Adc_Adcdig_aDmaChannels[Unit]; /** @violates @ref Adc_Adcdig_Irq_c_REF_9 cast from unsigned int to pointer */ /** @violates @ref Adc_Adcdig_Irq_c_REF_3 A cast should not be performed between a pointer type and an integral type. */ pTcdAddress = (Mcl_DmaTcdType*)Mcl_DmaGetChannelTcdAddress(DmaChannel); SchM_Enter_Adc_ADC_EXCLUSIVE_AREA_13(); /** @violates @ref Adc_Adcdig_Irq_c_REF_5 Array indexing shall be the only allowed form of pointer arithmetic */ if(Adc_pCfgPtr->pGroups[Group].NumSamples == Mcl_DmaTcdGetIterCount(pTcdAddress)) { Adc_aGroupStatus[Group].eConversion = ADC_STREAM_COMPLETED; } else { Adc_aGroupStatus[Group].eConversion = ADC_COMPLETED; } Adc_Adcdig_DmaCheckNotification(Group); } else #endif /* (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING) */ to #if (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING) /** @violates @ref Adc_Adcdig_Irq_c_REF_5 Array indexing shall be the only allowed form of pointer arithmetic */ if ((boolean) TRUE == Adc_pCfgPtr->pGroups[Group].bAdcDoubleBuffering) { DmaChannel = Adc_Adcdig_aDmaChannels[Unit]; /** @violates @ref Adc_Adcdig_Irq_c_REF_9 cast from unsigned int to pointer */ /** @violates @ref Adc_Adcdig_Irq_c_REF_3 A cast should not be performed between a pointer type and an integral type. */ pTcdAddress = (Mcl_DmaTcdType*)Mcl_DmaGetChannelTcdAddress(DmaChannel); SchM_Enter_Adc_ADC_EXCLUSIVE_AREA_13(); /** @violates @ref Adc_Adcdig_Irq_c_REF_5 Array indexing shall be the only allowed form of pointer arithmetic */ if(Adc_pCfgPtr->pGroups[Group].NumSamples == Mcl_DmaTcdGetIterCount(pTcdAddress)) { Adc_aGroupStatus[Group].eConversion = ADC_STREAM_COMPLETED; } else { Adc_aGroupStatus[Group].eConversion = ADC_COMPLETED; } Adc_Adcdig_DmaCheckNotification(Group); #if ((CACHE_INVALIDATE_MACROS

Table continues on the next page...

ID	Subtype	Headline and Description
		=STD_ON)&&(MCAL_CACHE_RUNTIME_MNGMNT)&&(MCL_LMEM_ENABLE_CACHE_API=STD_ON)) Mcl_CacheClear(MCL_LMEM_CACHE_ALL); #endif } else #endif /* (STD_ON == ADC_ENABLE_DOUBLE_BUFFERING) */','Change in Adc_Adcdig_Irq.c
ENGR00391466	NewWork	[ADC] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.','improve including mcal header structure
ENGR00389860	Defect	[ADC] Inconsistent guarding of IPV functions Adc_Adcdig_StartInjectedConversion, Adc_Adcdig_SetJcmrRegisters, Adc_Adcdig_EndHardwareConv '-- Converted from JIRA ticket MCAL-15998 -- Detailed description (how to reproduce it): 1. In the Adc_Adcdig.c the Adc_Adcdig_StartInjectedConversion is guarded by both ADC_ENABLE_START_STOP_GROUP_API and ADC_SOFTWARE_INJECTED_CONVERSIONS_USED, this is correct, but: In the Adc_Adcdig.h and Adc_Ipw.h, it's guarded by only ADC_ENABLE_START_STOP_GROUP_API. 2. In the Adc-Adcdig.h, the function Adc_Adcdig_SetJcmrRegisters is guarded by ADC_SETCHANNEL_API, but it's definition is not guarded by ADC_SETCHANNEL_API in the Adc_Adcdig.c 3. In the Adc_Adcdig_Irq.h, the function Adc_Adcdig_EndHardwareConv is guarded by ADC_UNIT_X_ISR_USED, but it's extern in the Adc_Bctu_Irq.c is not guarded by ADC_UNIT_X_ISR_USED Preconditions: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Misra errors are generated in the report, related to this inconsistency Expected behavior: No Misra errors to begenerated in the report, related to this inconsistency 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,ENGR00391359
ENGR00388552	Defect	[ADC] Incorrect MemMap start and stop section defines 'Problem detailed description (how to reproduce it): In Adc_Adcdig.h and Adc_Bctu.h, the define ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED is started but ADC_STOP_SEC_VAR_INIT_UNSPECIFIED is closed, making an error: ADC_STOP_SEC_VAR_INIT_UNSPECIFIED: This section has not been previously started! Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: compile error Expected behavior: Define ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Change ADC_STOP_SEC_VAR_INIT_UNSPECIFIED to ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED",'N/A,ENGR00388825
ENGR00388026	Defect	[ADC] Interrupt flags are not cleared after stopping a continuous group 'Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): In case of stopping continuous groups , the interrupt flag might not be cleared: Adc_Ipw_StopCurrentConversion cleares flag before stopping the group, in case of continuous group, hardware still active then another flag can be raised. However, when the next group is started, any pending flag is cleared first. 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): ISR flag is active after a group is stopped. This can cause a spurious ADC interrupt. Expected behavior: no ISR flag should be active after stopping a group 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.) 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): Clear the flags after stopping the continuous conversions, not before.","'Move the clearing interrupt flags under the stopping conversion in Adc_Adc12bsarv2_StopCurrentConversion, Adc_Adcdig_StopCurrentConversion
ENGR00389928	Defect	[ADC] Issue found by methodology configuration test for Calypso 4.2 RTM 100 '-- Converted from JIRA ticket MCAL-15104 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): -When implementing the methodology test for Calypso 4.2 RTM 100, I have found some issues: + Generate failure when enabling ConvTimeOnce and the node AdcNormalConvTimings is not exist (do not click on it), because of below condition: [!IF "((../../../NonAutosar/AdcEnableDualClockMode = 'true') or (../../../ NonAutosar/AdcConvTimeOnce = 'true') and (node:exists(AdcNormalConvTimings)))"![!// /**< @brief Normal Conversion time - CTR0 */ (Adc_ConversionTimeType) (ADCDIG_INPSAMP_VALUE_U32([!"num:i(AdcNormalConvTimings/ AdcSamplingDurationNormal)"!])), [!ELSE!][!// (Adc_ConversionTimeType)0x0, [!ENDIF!][!/ And the default value 0x00 is incorrect. + Generate failure when enable the BCTU DMA transfer, config 2 units and only config DMA for BCTU_0 in MCL module, because of below conditons: [!FOR "Unit" = "0" TO "num:i($maxUnit)"![!// [!VAR "mcl_Id"="0"![!// [!VAR "foundDma"="0"![!// [!VAR "UnitExits" = "0"![!// [!LOOP "AdcConfigSet/BCTUHwUnit/*[1]/BCTU_InputTrigger/*/ AdcChannelTriggered/*"![!// [!IF "ADCHWUNIT = concat('ADC',string(num:i($Unit)))"![!// [!VAR "UnitExits"="$UnitExits+1"![!// [!BREAK!][!// [!ENDIF!][!// [!ENDLOOP!][!// [!IF "AdcConfigSet/BCTUHwUnit/*[1]/BCTUDMAChannelEnable"![!// [!LOOP "node:refs('ASPathDataOfSchema:/TS_T2D35M10I0R0/Mcl')/ MclConfigSet/*[1]/DMAChannel/*"![!// [!IF "$UnitExits = 1"![!// [!VAR "temp"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 16)))))"![!// [!IF "$temp = concat('BCTU_',string(num:i($Unit)))"![!// [!VAR "foundDma"="1"![!// [!VAR "mcl_Id"="string(MclDMAChannelId)"![!// [!ENDIF![!//

Table continues on the next page...

ID	Subtype	Headline and Description
		[!ENDIF!][!// [!ENDLOOP!][!// [!IF "$UnitExits = 1"!][!// [!IF "$foundDma = 0"!][!// [!ERROR "No DMA channels enabled and configured in the MCL Plugin for transfer conversion results for BCTU_ADC. Enable them into MCL Plugin."!][!// [!ENDIF!][!// + Build failure: When DMA used, MHT group enable, no Interrupt is enabled in Adc Interrupt. -Function Adc_Adcdig_EndMultipleCtuConv is built. -Function Adc_Adcdig_RecordResult is not built But in Adc_Adcdig_EndMultipleCtuConv we call Adc_Adcdig_RecordResult >> build failed. The error log: STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Adcdig_Irq.c", line 2116: error #223-D: STDERR: function Adc_Adcdig_RecordResult declared implicitly Test Case ID (internal TC that caught the bug) - optional Adc_TS_Eq_Cot_01','N/A
ENGR00391171	Defect	[ADC] Issue found by methodology configuration test for Calypso 4.2 RTM 100 '-- Converted from JIRA ticket MCAL-15104 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): -When implementing the methodology test for Calypso 4.2 RTM 100, I have found some issues: + Generate failure when enabling ConvTimeOnce and the node AdcNormalConvTimings is not exist (do not click on it), because of below condition: [!IF "((../../../NonAutosar/AdcEnableDualClockMode = 'true') or (../../../NonAutosar/AdcConvTimeOnce = 'true') and (node:exists(AdcNormalConvTimings)))"!][!// /**< @brief Normal Conversion time - CTR0 */ (Adc_ConversionTimeType) (ADCDIG_INPSAMP_VALUE_U32([!"num:i(AdcNormalConvTimings/AdcSamplingDurationNormal)"!])), [!ELSE!][!// (Adc_ConversionTimeType)0x0, [!ENDIF!][!/ And the default value 0x00 is incorrect. + Generate failure when enable the BCTU DMA transfer, config 2 units and only config DMA for BCTU_0 in MCL module, because of below conditons: [!FOR "Unit" = "0" TO "num:i($maxUnit)"!][!// [!VAR "mcl_Id"="0"!][!// [!VAR "foundDma"="0"!][!// [!VAR "UnitExits" = "0"!][!// [!LOOP "AdcConfigSet/BCTUHwUnit/*[1]/BCTU_InputTrigger/*/AdcChannelTriggered/*"!][!// [!IF "ADCHWUNIT = concat('ADC',string(num:i($Unit)))"!][!// [!VAR "UnitExits"="$UnitExits+1"!][!// [!BREAK!][!// [!ENDIF!][!// [!ENDLOOP!][!// [!IF "AdcConfigSet/BCTUHwUnit/*[1]/BCTUDMAChannelEnable"!][!// [!LOOP "node:refs('ASPathDataOfSchema:/TS_T2D35M10I0R0/Mcl')/

Table continues on the next page...

ID	Subtype	Headline and Description
		MclConfigSet/*[1]/DMAChannel/*"![!// [!IF "$UnitExits = 1"![!// [!VAR "temp"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 16)))))"![!// [!IF "$temp = concat('BCTU_',string(num:i($Unit)))"![!// [!VAR "foundDma"="1"![!// [!VAR "mcl_Id"="string(MclDMAChannelId)"![!// [!ENDIF![!// [!ENDIF![!// [!ENDLOOP![!// [!IF "$UnitExits = 1"![!// [!IF "$foundDma = 0"![!// [!ERROR "No DMA channels enabled and configured in the MCL Plugin for transfer conversion results for BCTU_ADC. Enable them into MCL Plugin."![!// [!ENDIF![!// + Build failure: When DMA used, MHT group enable, no Interrupt is enabled in Adc Interrupt. -Function Adc_Adcdig_EndMultipleCtuConv is built. -Function Adc_Adcdig_RecordResult is not built But in Adc_Adcdig_EndMultipleCtuConv we call Adc_Adcdig_RecordResult >> build failed. The error log: STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/Adc_Adcdig_Irq.c", line 2116: error #223-D: STDERR: function Adc_Adcdig_RecordResult declared implicitly Test Case ID (internal TC that caught the bug) - optional Adc_TS_Eq_Cot_01",'N/A
ENGR00389714	Defect	[ADC] Remove sign file for Adc_VersionCheck_Inc.m in Adc.mak '-- Converted from JIRA ticket MCAL-15044 -- generate_PB\Adc_VersionCheck_Inc.m this file is not in the plugin aymore but it's still in the list of signed files",'N/A
ENGR00390686	Defect	[ADC] Rename Adc_UnitStatus to Adc_aUnitStatus in Adc_Bctu_Irq.c '-- Converted from JIRA ticket MCAL-17165 -- Detailed description (how to reproduce it): Adc_UnitStatus renamed to Adc_aUnitStatus. But in Adc_Bctu_Irq.c it's not updated. So it can raise fail at build when multi hardware trigger used. Preconditions: Multi hardware trigger used for group. 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
		[...]",'N/A
ENGR00392085	Defect	[ADC] Should not disable hardware in the loop '-- Converted from JIRA ticket MCAL-18049 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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",'N/A
ENGR00388938	NewWork	[ADC] Support for running from User Mode 'NewWork Description: 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A
ENGR00390823	Defect	[ADC] The function Adc_Adcdig_HwResultReadGroup alway returns E_NOT_OK if all channel are converted yet 'Problem detailed description (how to reproduce it): The Group 0 is configured as hw trigger. In the container AdcGroupDefinition: AN0, AN1, AN2, AN3, AN4. If we call Adc_Adcdig_HwResultReadGroup when: AN0, AN1 is converted. bit VALID is 1 AN2, AN3, AN4 isn't converted yet. Adc_Adcdig_HwResultReadGroup read VALID bit of AN0, AN1 and hardware resit the bits to 0 automatically. Next step we call Adc_Adcdig_HwResultReadGroup again. In this case VALID bit of AN0, AN1 is 0 and Adc_Adcdig_HwResultReadGroup always return E_NOT_OK even AN2, AN3, AN is converted. Suggestion:

Table continues on the next page...

ID	Subtype	Headline and Description
		We should use extra array to manage status of each channel.",'N/A
ENGR00390810	Defect	[ADC] The static variable should not be declared inside functions '-- Converted from JIRA ticket MCAL-17431 -- 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: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA 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): NA","'The static variable should not be declared inside functions Files modified: Adc.c",ENGR00391182
ENGR00389712	Defect	[ADC] Update Range for AdcPresamplingInternalSignal '-- Converted from JIRA ticket MCAL-15006 -- Update Range for AdcPresamplingInternalSignal as the description in 32.6.4.1 Presampling channel enable chapter in MPC5748G_RM_Rev5 (the attachment from MCAL-14911)",'N/A
ENGR00388791	Defect	[ADC] aAdc_HwQueueMaxDepth in Adc_Adcdig_MultiConfigType should be guarded 'Problem detailed description (how to reproduce it): aAdc_HwQueueMaxDepth element in Adc_Adcdig_MultiConfigType only be used when (ADC_MULTIPLE_HARDWARE_TRIGGERS == STD_ON). So it should be guarded under (ADC_MULTIPLE_HARDWARE_TRIGGERS == STD_ON) to minimize the struct Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Trigger: N/A Observed behavior: N/A Expected behavior: Guard the aAdc_HwQueueMaxDepth and update the template file too. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): aAdc_HwQueueMaxDepth element in Adc_Adcdig_MultiConfigType only be used when (ADC_MULTIPLE_HARDWARE_TRIGGERS == STD_ON). So it should be guarded under (ADC_MULTIPLE_HARDWARE_TRIGGERS == STD_ON) to minimize the struct",'Adc_Adcdig_CfgEx.h
ENGR00391469	NewWork	[ALL/GENERAL]Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00391562	NewWork	[ALL/GENERAL]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

Table continues on the next page...

ID	Subtype	Headline and Description
		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: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA",'Improvements of running from User Mode
ENGR00390998	NewWork	[BASE] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00389875	Defect	[CAN] CanAbortOnlyOneMB option must depend on CanApiEnableMbAbort '-- Converted from JIRA ticket MCAL-16406 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]",'CanAbortOnlyOneMB option must depend on CanApiEnableMbAbort
ENGR00389800	Defect	[CAN] CanTableIDType must not accept Extended ID when CanExtendedIDSupport=OFF '-- Converted from JIRA ticket MCAL-16409 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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): [...]",'CanTableIDType must not accept Extended ID when CanExtendedIDSupport=OFF
ENGR00388418	Defect	[CAN] Can_pControlerDescriptors as Null in Can_FlexCan_InitVariables 'Problem detailed description (how to reproduce it): Line 3335, Can_FlexCan.c: Can_ControllerStatuses[u8CtrlIndex].u8CurrentBaudRateIndex = (uint8) (Can_pControlerDescriptors[u8CtrlIndex].u8DefaultBaudRateIndex); Got IVOR2 handler during process in Can_FlexCan_InitVariables() because Can_pControlerDescriptors as null ptr. Preconditions: 1. Call Can_Init(); 2. Precompile = ON 3. CanDevErrorDetection = OFF Test Case ID (internal TC that caught the defect) - optional Can_TC_0002 Trigger: n/a

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: IVOR2 Expected behavior: without IVOR2 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 Can.c, lin 551, 552: CanStatic_pCurrentConfig = &CanStatic_ConfigSet; CanStatic_pControlerDescriptors = CanStatic_pCurrentConfig->StaticControlerDescriptors;","'Line 3335, Can_FlexCan.c: Can_ControllerStatuses[u8CtrlIndex].u8CurrentBaudRateIndex = (uint8)(Can_pControlerDescriptors[u8CtrlIndex].u8DefaultBaudRateIndex); Got IVOR2 handler during process in Can_FlexCan_InitVariables() because Can_pControlerDescriptors as null ptr. Preconditions: Precompile = ON CanDevErrorDetection = OFF
ENGR00389359	Defect	[CAN] FLEXCAN_CBT[BTF] is not set even if CanControllerCbtEnable is true 'Problem detailed description (how to reproduce it): The customer configures CAN driver with CanControllerCbtEnable = true but the generated value of FLEXCAN_CBT register does not have BTF =1. Because of this, the CAN baud rate is not correct as their expectation, then the CAN communication does not work. Preconditions: CanControllerCbtEnable is set. The configuration of bit timming in CanControllerCBT different from which is used in classic configuration (for FLEXCAN_CTRL1). Observed behavior: CAN is not working. When can it be observed? (at configuration time, at runtime, at compile time?) at runtime Expected behavior: CAN is working Proposed solution (Optional): In the template code (Can_Cfg.c/Can_PBCfg.c) to generate the configuration value of FLEXCAN_CBT register, add the code to generate BTF bit. e.g: Replace [!IF "ecu:get('Can.CanConfigSet.CBTSupport')='STD_ON'"!] #if (CAN_CBT_ENABLE == STD_ON) { [!INDENT "12"!] [!IF "(node:value(CanControllerCBT/CanControllerCbtEnable)) = 'true'"!] (uint32)((uint32)1U<<FLEXCAN_CBT_OFFSET_U8), (uint32)[!"CanControllerCBT/CanControllerCbtBaudRate"!]U, /* [!"CanControllerCBT/CanControllerCbtBaudRate"!]kbps baud rate */ (uint32)((uint32)[!"num:i((CanControllerCBT/CanControllerCbtPrescaller)-1)"!]U << FLEXCAN_FD_PRESDIV_CBT_OFFSET_U8) | /* CAN FD prescaller */ by [!IF "ecu:get('Can.CanConfigSet.CBTSupport')='STD_ON'"!] #if (CAN_CBT_ENABLE == STD_ON) { [!INDENT "12"!]

Table continues on the next page...

ID	Subtype	Headline and Description
		[!IF "(node:value(CanControllerCBT/CanControllerCbtEnable)) = 'true'"!] (uint32)((uint32)1U<<FLEXCAN_CBT_OFFSET_U8), (uint32)[!"CanControllerCBT/CanControllerCbtBaudRate"!]U, /* [!"CanControllerCBT/CanControllerCbtBaudRate"!]kbps baud rate */ (uint32)((uint32)[!"num:i((CanControllerCBT/CanControllerCbtPrescaller)-1)"!]U << FLEXCAN_FD_PRESDIV_CBT_OFFSET_U8) | /* CAN FD prescaller */ [!IF "(node:value(CanControllerCBT/CanControllerCbtEnable)) = 'true'"!] (uint32)((uint32)1U<<FLEXCAN_CBT_OFFSET_U8) | [!ELSE!] (uint32)((uint32)0U<<FLEXCAN_CBT_OFFSET_U8) | [!ENDIF!]","'<Description of changes, e.g. Check was added to ... The function xyz was modified because ...> Files Modified: <e.g. Adc_Irq.c > <...>
ENGR00388687	NewWork	[CAN] Improve generic source to support multiple platforms 'NewWork Description: - With different file structure in Can modules: all of generate files, configuration files... are generic - Some actions for improvement, to work on multiple platforms. Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Improvement",'just for improvement.
ENGR00390907	Defect	[CAN] Incorrect range for CanRxProcessing/CanTxProcessing/ CanWakeupProcessing/CanBusoffProcessing '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.",'N/A

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00388806	NewWork	[CAN] Support for running from User Mode 'NewWork Description: 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 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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'add to support running in usermode
ENGR00388805	Defect	[CAN] The CAN_FDCTRLis 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. Setting the TDCEN bit enables the Transceiver Delay Compensation feature for transmission at high baud-rate in CAN-FD mode. Because this bit is not set, the feature is disabled. Consequently, depend on the transceiver the bit error may occur and the transmission cannot be done when the driver transmits data at high baud-rate in CAN-FD mode. Preconditions: CanControllerTxBitRateSwitch is checked. Test Case ID (internal TC that caught the defect) - optional NA 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));",' Can_Flexcan.c
ENGR00388865	Defect	[CAN] The UM contains M4 macros 'Problem detailed description (how to reproduce it): Can AUTOSAR_MCAL_CAN_UM.pdf: a lot of M4 macros in the documentation Preconditions: [NA] Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		[NA] Trigger: [NA] Observed behavior: Can AUTOSAR_MCAL_CAN_UM.pdf: a lot of M4 macros in the documentation Expected behavior: The M4 macros should be replaced by right values. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [replace M4 macro by CAN]",'N/A
ENGR00388733	NewWork	[CAN] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.",'N/A
ENGR00389953	Defect	[CAN] When FD is activated without BRS in configuration we still need to configure the second baudrate fields even not needed '-- Converted from JIRA ticket MCAL-16335 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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:

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): [...]",'When FD is activated without BRS in configuration we still need to configure the second baudrate fields even not needed
ENGR00388520	Defect	[CAN] Wrong reference to non-autosar parameter 'Problem detailed description (how to reproduce it): Not impact to customer application, just update reference with correctly ref of autosar container. 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> Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: wrong reference Expected behavior: correct reference Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): 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> Solution: Replace in Can.xdm: <a:v>ASPathDataOfSchema:/AUTOSAR/EcucDefs/Can/CanConfigSet/CanController/CanRAMBlock</a:v> with <a:v>ASPathDataOfSchema:/M4_XDM_AR_PKG_NAME/Can/CanConfigSet/CanController/CanRAMBlock</a:v>",'update wrong autosar reference
ENGR00389916	Defect	[DIO] "<" character must not be contained in the attribute "true" which associated with an element type "null" '-- Converted from JIRA ticket MCAL-15959 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ

Table continues on the next page...

ID	Subtype	Headline and Description
		currently. 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): [...]",'N/A
ENGR00389951	Defect	[DIO] Correct the description of Dio Reverse Port Bits '-- Converted from JIRA ticket MCAL-16299 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. 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.",'N/A
ENGR00389751	Defect	[DIO] Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2 '-- Converted from JIRA ticket MCAL-14877 -- The attachment 'AUTOSAR_MCAL_DIO_Checklist_for_Code_Review_Intermediate.xls'

Table continues on the next page...

ID	Subtype	Headline and Description
		exceeded 50 characters filename CQ limit, it was therefore reduced. Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2",'N/A
ENGR00387612	Defect	 [DIO] 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: #define SIUL2_REVERSE_BITS_NUMBER ((Dio_PortLevelType) ((sizeof(Dio_PortLevelType) << 0x3U) - 1U)) 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 #define SIUL2_REVERSE_BITS_NUMBER ((Dio_PortLevelType) ((sizeof(Dio_PortLevelType) << 0x3U) - 1U)) To #define SIUL2_REVERSE_BITS_NUMBER ((Dio_PortLevelType) ((((uint8)sizeof(Dio_PortLevelType)) << 0x3U) - 1U))",'Fix misra violations
ENGR00389803	Defect	 [DIO] Missing extern Dio_ConfigPC in link time mode '-- Converted from JIRA ticket MCAL-14985 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. In the Dio.c, the variable Dio_ConfigPC (Dio configuration structure extern) is guarded under DIO_PRECOMPILE_SUPPORT: #if defined(DIO_PRECOMPILE_SUPPORT) #define DIO_START_SEC_CONFIG_DATA_UNSPECIFIED /** @violates @ref Dio_C_REF_2 Precautions to prevent the contents of a header file being included twice */ /** @violates @ref Dio_C_REF_4 Only preprocessor statements and comments before '#include'.*/ #include "Dio_MemMap.h" /* Local Variable for storing the generated Dio configuration */ /** @violates @ref Dio_C_REF_7 Objects shall be defined at block scope if they are only accessed from within a single function.*/ extern CONST(Dio_ConfigType, DIO_CONST) Dio_ConfigPC;

Table continues on the next page...

ID	Subtype	Headline and Description
		#define DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED /** @violates @ref Dio_C_REF_2 Precautions to prevent the contents of a header file being included twice */ /** @violates @ref Dio_C_REF_4 Only preprocessor statements and comments before '#include'.*/ #include "Dio_MemMap.h" #endif So in case linking time is selected, the code under DioDevErrorDetect, DioFlipChannelApi and DioVersionInfoApi which are calling to Dio_ConfigPC cannot see the the variable due to it's not extern in Dio.c.','N/A
ENGR00388734	NewWork	[DIO] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.','N/A
ENGR00389952	Defect	[ETH] Duplicate code line '-- Converted from JIRA ticket MCAL-16403 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): This issue happen in Eth_Irq.c, line 264 and 276 are duplicate in case: FALSE = Eth_Ipw_IsRxInterruptEnabled(u8CtrlIdx) (line: 249). Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Duplicate code

Table continues on the next page...

ID	Subtype	Headline and Description
		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): RX interrupt flag will be cleared anyway -> remove duplicate code in case FALSE = Eth_Ipw_IsRxInterruptEnabled(u8CtrlIdx)",'Eth_Irq.c
ENGR00389868	Defect	[ETH] EthGlobalTime is incremented just after Eth_SetGlobalTime (before next timer occurred) '-- Converted from JIRA ticket MCAL-15955 -- These ConfigAffected: IMXVS4_4.2, SC667349_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): If the timer occurred during setting time, driver will continue setup the time without clear the interrupt flag. Therefore, after exit the critical section, the second part of time will be increment 1 cause incorrect time set by driver. Preconditions: -Use Eth Timestamp -Interrupt occurred in a specific time (calling Eth_SetGlobalTime) Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Seconds part of timer is incremented 1 value just after exit from Eth_SetGlobalTime Expected behavior: Seconds will be same when exit from Eth_SetGlobalTime Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Clear the interrupt after set the global time to not process the previous wrap. Update in file Eth_Enet.c, in function Eth_Enet_SetGlobalTime, after set the seconds part of time.",'Eth_Enet.c
ENGR00389723	Defect	[ETH] Improve code to select controller exactly to be configured '-- Converted from JIRA ticket MCAL-14863 -- when ETH_MAXCTRLS_SUPPORTED is greater than 1, If user only use controller 1 in configuration, but Eth_Init function use other controller (controller 0) Proposed Solution: correct condition in for loop of Eth_Init function to select controller exactly to be configured","'Eth_Cfg.h Eth.h
ENGR00388527	NewWork	[ETH] Support for running from User Mode 'NewWork Description: 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

Table continues on the next page...

ID	Subtype	Headline and Description
		to activate the specific implementation for non-privileged mode. By default, '<Mdl>EnableUserModeSupport' field shall be disabled 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 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. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A,ENGR00391374 ENGR00391375
ENGR00389552	Defect	[ETH] The data reception is stopped after re-activated by Eth_SetControllerMode 'Problem detailed description (how to reproduce it): The Eth reception was totally not working after this scenario: - Eth channel is in Full communication state and Eth working properly. - Channel is requested to go to No communication, so the controller mode is changed to DOWN. - Channel is requested to go to Full communication, so the controller mode is changed to ACTIVE. CE's comment: The test scenario: 1. Initialize Ethernet stack 2. Eth_Init 3. Eth_ControllerInit 4. Eth_SetControllerMode(ETH_MODE_ACTIVE) 5. Do some communication 6. Eth_SetControllerMode(CONTROLLER, ETH_MODE_DOWN); 7. Eth_SetControllerMode(ETH_MODE_ACTIVE) 8. The driver cannot receive data After the function Eth_SetControllerMode is called to put the controller into DOWN state, the ENETx_ECR[ETHEREN] is cleared by the driver and the ENETx_MRBR is also cleared by hardware after that. However, when the function Eth_SetControllerMode is called to put the controller into the ACTIVE state for full-communication, the register ENETx_MRBR is not re-configured and it is still 0, this is improper configuration of ENETx_MRBR register. It means the maximum receive buffer size is incorrectly configured to 0 byte. Consequently the ENET could not receive the data anymore. For more details about the debugging of this defect, I added the following code after getting the buffer length (u16BufLength = ETH_CFG_CTRLRXBUFLENBYTE(Eth_CtrlCfg, u8CtrlIdx);) in Eth_Enet_ConfigureRxBuffers function and it worked fine. /* Init the active Rx buffer descriptor pointer - the BDs are reset during controller reset or disablement */ /** @violates @ref Eth_Enet_c_REF_7 MISRA rule 11.1 */ REG_WRITE32(Eth_u32BaseAddr[u8CtrlIdx] + ENET_MRBR_ADDR16, u16BufLength); Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The driver could not receive data after re-activating the controller. Expected behavior: The driver could receive data after re-activating the controller. 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
		NA",'Eth_Enet.c
ENGR00389885	Defect	[ETH] Wrong EthInterPacketGap range value '-- Converted from JIRA ticket MCAL-16369 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): According S32K14x Series Reference Manual, Rev. 3, 03/2017 (page: 1762) describe " thInterPacketGap (IPG bits in ENET_TIPG register): Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12." But in current implementation for S32K14X, the range for EthInterPacketGap is 8 to 27 instead of 8-26 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","'Eth.mak Eth.xdm Eth_Checks.m
ENGR00389887	Defect	[ETH] Wrong behavior of Eth_UpdatePhysAddrFilter() API '-- Converted from JIRA ticket MCAL-16379 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): According requirement SWS_Eth_00147: "If the physical source address (MAC address) is set to 00:00:00: 00:00:00, this shall reduce the filter to the controllers unique unicast MAC address and end promiscuous mode if it was turned on." But in current implementation, driver did not end promiscuous mode when it was was turned on. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Wrong behavior of Eth_UpdatePhysAddrFilter() API Expected behavior: If the physical source address (MAC address) is set to 00:00:00: 00:00:00 , driver will end promiscuous. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): If the physical source address (MAC address) is set to 00:00:00: 00:00:00 , driver will end promiscuous when it was turned

Table continues on the next page...

ID	Subtype	Headline and Description
		on",'Eth_Enet.c,ENGR00392029
ENGR00388742	NewWork	[FEE] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.",'N/A
ENGR00387843	Defect	[FEE] Warnings in Tresos when loading modules 'Problem detailed description (how to reproduce it): When loading Fee module in Tresos, we get the following warnings: Ignoring attribute "IMPLEMENTATIONCONFIGCLASS" of "/TS_T2D30M20I0R0/Fee/FeeBlockConfiguration": Unknown attribute Ignoring attribute "IMPLEMENTATIONCONFIGCLASS" of "/TS_T2D30M20I0R0/Fee/FeeClusterGroup": Unknown attribute. Ignoring attribute "IMPLEMENTATIONCONFIGCLASS" of "/TS_T2D30M20I0R0/Fee/FeeClusterGroup/FeeCluster": Unknown attribute. Ignoring attribute "IMPLEMENTATIONCONFIGCLASS" of "/TS_T2D30M20I0R0/Fee/FeeClusterGroup/FeeCluster/FeeSector": Unknown attribute. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Tresos warnings Expected behavior: No Tresos warnings

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",'N/A
ENGR00388038	Defect	[FLS] Follow up : Fix Code review checklist findings in Calypso 'Initial Description: Fls driver violates code rules: SMCALRule 5.24 ex: VAR(boolean, FLS_VAR) bRetVal; Problem detailed description (how to reproduce it): Please refer to code reviewcheckist at the attachments Preconditions: N/A Observed behavior: Please refer to code reviewcheckist at the attachments When can it be observed? (at configuration time, at runtime, at compile time?) In the review code phase Expected behavior: Investigate those violations, Fix those violated rule if needed Reported release baseline: BLN_FEE_SMCAL_4.0_MPC574XG_01.04.00 Proposed solution (Optional): NewWork Classification: (internal task, improvement, feature request) NA In case for feature request, does a TWG ticket exist (cloned or linked)? TBC (Yes/No) NewWork Description: NA Expected behavior: Fix rule: SMCALRule 5.2 Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)","'<Description of changes, e.g. Check was added to ... The function xyz was modified because ...> Files Modified: <e.g. Adc_Irq.c > <...>
ENGR00391406	NewWork	[FLS] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the

Table continues on the next page...

ID	Subtype	Headline and Description
		prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00390918	Defect	[FLS] Incorrect implementation for FLS_TIMEOUT_HANDLING check '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)",'N/A
ENGR00388570	NewWork	[FLS] Support for running from User Mode 'NewWork Description: 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A,ENGR00388572 ENGR00389653
ENGR00388743	NewWork	[FLS] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.','N/A
ENGR00390739	Defect	[FR] Build failed due to missing compiler switch '-- Converted from JIRA ticket MCAL-17210 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. No valid config left, ALL/GENERAL used instead. Detailed description (how to reproduce it): We got an build issue when build code with User Mode support: STDERR:"e:/local_02/output/projects/ar_epd_fr_ghs/tresos/plugins/Fr_TS_T31D17M10I0R0/src/Fr.c", line 594: error #223-D: STDERR: function Fr_IPW_SetUserAccessAllowed declared implicitly STDERR: Fr_IPW_SetUserAccessAllowed(); STDERR: ^ STDERR: Because the function: Fr_IPW_SetUserAccessAllowed is only defined in case: FR_ENABLE_USER_MODE_SUPPORT=ON and FR_CONFIG_REG_PROT defined before. so the code (line 594) must be change to: #if (STD_ON == FR_ENABLE_USER_MODE_SUPPORT) #ifdef FR_CONFIG_REG_PROT Fr_IPW_SetUserAccessAllowed(); #endif #endif/*FR_ENABLE_USER_MODE_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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): [...]",'N/A
ENGR00390764	Defect	[FR] Driver does not have enough memory for buffer '-- Converted from JIRA ticket MCAL-17287 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): When sharing MB is enable, it is observed that the memory allocate for FR driver is smaller than expected. Therefore, during synchronization, Fr access to in-allowed memory area. This cause test fail. Preconditions: Enable Re-config Lpdu Test Case ID (internal TC that caught the defect) - optional Fr_TS_09_EXT Observed behavior: Memory is smaller than required Expected behavior: Fr access to memory outside of grand area. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Increase the memory allocated for Fr Buffer.",'N/A
ENGR00389866	Defect	[FR] FR_E_CTRL_TESTRESULT is not configurable '-- Converted from JIRA ticket MCAL-16263 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): The container FrControllerDemEventParameterRefs and the ref FR_E_CTRL_TESTRESULT have the attribute READONLY set to true making them not possible to configure. This issue happens when starting a new configuration on Tresos. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: FR_E_CTRL_TESTRESULT is not configurable Expected behavior: FR_E_CTRL_TESTRESULT should be configurable Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove READONLY attribute in container FrControllerDemEventParameterRefs",'N/A
ENGR00390612	Defect	[FR] Fix Misra Errors for Treerunner '-- Converted from JIRA ticket MCAL-17090 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ

Table continues on the next page...

ID	Subtype	Headline and Description
		currently. Detailed description (how to reproduce it): When running misra report for Fr the following errors are detected: Error(s) File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_02\plugins\Fr_TS_T31D17M10I0R0\src\Fr_Flexray.c", line 2933, Violates [MISRA 2004 Rule 11.3, advisory]: no relevant comment was found for MISRA violation on this line. File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_02\plugins\Fr_TS_T31D17M10I0R0\src\Fr_Flexray.c", line 4008, Violates MISRA 2004 Rule 17.4: no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_02\plugins\Fr_TS_T31D17M10I0R0\src\Fr_Flexray.c", line 4008, Violates MISRA 2004 Rule 12.4: no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_02\plugins\Fr_TS_T31D17M10I0R0\src\Fr_Flexray.c", line 4009, Violates MISRA 2004 Rule 17.4: no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_21.0.0_b160607_0933_02\plugins\Fr_TS_T31D17M10I0R0\src\Fr_Flexray.c", line 4009, Violates MISRA 2004 Rule 12.4: no MISRA violation comment was found (maybe wrong format is used). Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Observed behavior: [...] Expected behavior: No misra errors should be present 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
ENGR00391689	NewWork	[FR] Support for running from User Mode '-- Converted from JIRA ticket MCAL-15023 -- NewWork Description: 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 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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A,ENGR00391691
ENGR00389562	Defect	[FR] The drivers should accept the received data on the buffer assigned for channel AB if the slot status shows a valid frame (A/B) '-- Converted from JIRA ticket MCAL-16720 --

Table continues on the next page...

ID	Subtype	Headline and Description
		Git Branch: bugfix/MCAL-16720-fr-the-drivers-should-accept-the Problem detailed description (how to reproduce it): Below is report from our customer. I am currently working with Freescale MCAL Flexray driver for MPC5777M. I am having a problem with data reception in "FRIF_CHANNEL_AB" message buffer. Please see below detailed explanation. - What we expect: When setup the receive buffer in both channel AB that data can be received on either channel A or channel B. It means if I configure the receiver to receive data on channel AB and transmitter send data in only channel A, data shall be received successfully. - What actually happened: Data cannot be received on receiver with above setup. - What we analyzed so far: see that when configured "FRIF_CHANNEL_AB" for message buffer, the routine Fr_Flexray_ReceiveRxLPdu can reach the channel checking, but cannot pass the stringent checking for channel AB. I configured receive buffer to channel AB, slot 12, generated Lpdu Id for that slot is 4. CE's comment: According to Table 54-17. Channel Assignment Description in the reference manual (Rev4) and the Flexray specification, the buffer which is assigned for both channel A&B could receive data on either channel A or channel B in static slot. The following AUTOSAR requirements (FR603 and FR604) say that the received data is accepted if there is a valid frame showed in the slot status. [FR604]If stringent check is disabled by configuration parameter (FrRxStringentCheck is false), then received data is accepted if the SlotStatus shows a valid frame (vSS!ValidFrame). Otherwise FR_NOT_RECEIVED is written to Fr_RxLPduStatusPtr and 0 is written to Fr_LsduLengthPtr.() [FR603] If stringent check is enabled by configuration parameter (FrRxStringentCheck is true), then received data is accepted only if the SlotStatus shows a valid frame (vSS!ValidFrame) and there was no single SlotStatus error bit set (vSS!SyntaxError, vSS!ContentError, vSS!Bviolation). Otherwise FR_NOT_RECEIVED is written to Fr_RxLPduStatusPtr and 0 is written to Fr_LsduLengthPtr.() Currently, the driver only accepts the received data if there are two frames received on both channels in the slot. That may increase the liability or safety as any problem either a channel Aor channel B, the data is rejected but it loses the level of fault tolerance which the dual channel is designed for. Therefore, the Flexray driver shall be fixed to be aligned with Flexray protocol, Autosar specification and also hardware manual. The current implementation may be kept as an optional feature (liability & safety enhancement feature). In order to fix the issue, the line 3672 and 3401 in Fr_Flexray.c of MPC5777M MCAL4.0 RTM1.0.1 should be changed to: For Static slot: if((0x0080U == (pheaderMBPtr_12[3] & FLEXRAY_STRINGENT_MASK_A))| | (0x8000U == (pheaderMBPtr_12[3] & FLEXRAY_STRINGENT_MASK_B))) For Dynamic slot: if(0x0080U == (pheaderMBPtr_12[3] & FLEXRAY_STRINGENT_MASK_A)) Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The frame cannot be received due to the incorrectly checking of status slot description. Expected behavior: The frame could be received

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): [...]",'The drivers should accept the received data on the buffer assigned for channel AB if the slot status shows a valid frame (A/B)
ENGR00390384	Defect	 [FR] wrong allocation of memory sections '-- Converted from JIRA ticket MCAL-16977 -- These ConfigAffected: IMXVS4_4.2 were removed as they are invalid in CQ currently. No valid config left, ALL/GENERAL used instead. Detailed description (how to reproduce it): It happens in Fr.h line: 657, Fr_pCfgPtr must be set in INIT_UNSPECIFIED section instead of NO_INIT_UNSPECIFIED section because it is initialized before in Fr.c. This issue make build code failed ith IAR compiler. 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): [...]",'N/A
ENGR00391678	NewWork	 [GPT, ICU, MCU, OCU, PWM][MCL] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl

Table continues on the next page...

ID	Subtype	Headline and Description
		which affecting in customer raising.",'improve including mcal header structure
ENGR00387878	NewWork	[GPT] Add support Gpt_ChangeNextTimeoutValue for eMios 'NewWork Description: Add support Gpt_ChangeNextTimeoutValue for eMios platform, use SAOC mode. Requirement ID: PR-MCAL-3307.gpt 1.Service name: Gpt_ChangeNextTimeoutValue Syntax: void Gpt_ChangeNextTimeoutValue(Gpt_ChannelType Channel, Gpt_ValueType Value) Service ID[hex]: 0x0F Sync/Async: Synchronous Reentrancy: Reentrant Parameters (in): Channel - Numeric identifier of the GPT channel. Value ? Target time in number of ticks. Parameters (inout): None Parameters (out): None Return value: None Description: Change the compare value of a running timer channel. 2.The function Gpt_ChangeNextTimeoutValue shall be reentrant, if the timer channels used in concurrent calls are different. 3.If development error detection for the GPT module is enabled: If the parameter Channel is invalid (not within the range specified by configuration), the function Gpt_ChangeNextTimeoutValue shall raise the error GPT_E_PARAM_CHANNEL. 4.If development error detection for the GPT module is enabled:The function Gpt_ChangeNextTimeoutValue shall raise the error GPT_E_PARAM_VALUE if the parameter Value is "0" or not within the allowed range (exceeding the maximum timer resolution). 6.If development error detection for the GPT module is enabled: If the driver is not initialized, the function Gpt_ChangeNextTimeoutValue shall raise the error GPT_E_UNINIT. 7.If development error detection for the GPT module is enabled: If the function Gpt_ChangeNextTimeoutValue is called on a channel in state "not running", the function shall raise the error GPT_E_INVALID_CALL. 8.If development error detection for the GPT module is enabled: If the function Gpt_ChangeNextTimeoutValue is called on a channel configured as one-shot mode, the function shall raise the error GPT_E_PARAM_CHANNEL. 9.If development error detection for the GPT module is enabled: If the function Gpt_ChangeNextTimeoutValue is called on a channel which is not supported by hardware to do its behaviour, the function shall raise the error GPT_E_PARAM_CHANNEL Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'add new api function: Gpt_ChangeNextTimeoutValue,ENGR00388803 ENGR00388854
ENGR00390049	Defect	[GPT] Channel status change shall be made before to start the hardware 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		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):",'Channel status change shall be made before to start the hardware
ENGR00390853	NewWork	[GPT] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl

Table continues on the next page...

ID	Subtype	Headline and Description
		which affecting in customer raising.",'improve including mcal header structure,ENGR00391607 ENGR00392971
ENGR00389837	Defect	[GPT] Remove ERR_IPV_PIT_0002 not used errata workaround '-- Converted from JIRA ticket MCAL-15824 -- These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): As analyzed in errata ticket for RRS RTM2.0.0 the workaround ERR_IPV_PIT_0002 is not used. Also for RRU it was never used (base was not taking into consideration) Proposed solution (Optional): Remove unnecessary errata workaround in Gpt_Pit.c",'Remove ERR_IPV_PIT_0002 not used errata workaround
ENGR00389855	Defect	[ICU, GPT, PWM][MCL] Add channel resource conflict checking '-- Converted from JIRA ticket MCAL-16179 -- Detailed description (how to reproduce it): GPT,OCU,ICU,PWM can use a channel without ISR enable. Currently, eMios_Common.c and eTimer_Common.c only check the conflict for ISR used. It is needed to replace ISR conflict check with Channel conflict check. Eg. #if ((defined ICU_ETIMER_2_CH_5_ISR_USED) && (defined GPT_ETIMER_2_CH_5_ISR_USED)) #error "ICU and GPT resource conflict for ETIMER unified channel ETIMER_2_5" #endif Should be #if ((defined ICU_ETIMER_2_CH_5_USED) && (defined GPT_ETIMER_2_CH_5_USED)) #error "ICU and GPT resource conflict for ETIMER unified channel ETIMER_2_5" #endif The convention of channel used define is bellow : [MDL]_ETIMER_[eTimer Module]_CH_[eTime Channel]_USED where [MDL] is OCU,PWM,ICU,GPT Similar to eMios...tracking resource channel conflict between drivers which using emios: /* * @brief resource conflict check for EMIOS_0_CH_0 */ #if ((defined GPT_EMIOS_0_CH_0_USED) && (defined PWM_EMIOS_0_CH_0_USED)) #error "GPT and PWM resource conflict for EMIOS unified channel EMIOS_0_0" #endif #if ((defined ICU_EMIOS_0_CH_0_USED) && (defined PWM_EMIOS_0_CH_0_USED)) #error "ICU and PWM resource conflict for EMIOS unified channel EMIOS_0_0" #endif For working currently: Wrong appear during sharing EMios channel between drivers when only one

Table continues on the next page...

ID	Subtype	Headline and Description
		has control of the interrupt ==> at compiling time will return an error, for ex: #error "PWM and ICU resource conflict for EMIOS unified channel EMIOS_0_0" Preconditions: NA 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): [...]",'update source code in eMios_Common.c file
ENGR00389500	Defect	[ICU, PORT] Icu_Siul2_SetUserAccessAllowed() uses PORT_CODE instead of ICU_CODE '-- Converted from JIRA ticket MCAL-15084 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): Siul2 Icu_Siul2_SetUserAccessAllowed() uses PORT_CODE instead of ICU_CODE in definition Preconditions: none Test Case ID (internal TC that caught the bug) - optional NA Trigger (not applicable in case of new features): use ICU driver Observed behavior (not applicable in case of new features): incorrect definition of data Expected behavior: corect use of ICU_DATA instead of PORT_DATA 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): use ICU_DATA in definition and declaration",'Icu_Siul2_SetUserAccessAllowed() uses PORT_CODE instead of ICU_CODE
ENGR00388523	NewWork	[ICU] Support for running from User Mode 'NewWork Description: 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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 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. Requirement source: cPRD (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):",'add to support usermode
ENGR00385082	Defect	[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):","'File modified: Icu_PluginMacros.m Updating at line 500, Replace of */!]EcuMConf_EcuMWakeupSource_[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/@name"!][!/* by */!]1<<[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/EcuMWakeupSourceId"!][!/*
ENGR00388745	NewWork	[ICU] User Manual for each driver should contain a "How To" configure chapter for advanced feateures

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.','N/A
ENGR00389881	Defect	[ICU] Validity of ICU...ChannelRef node is over-checked '-- Converted from JIRA ticket MCAL-16214 -- 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','Icu.xdm
ENGR00391171	Defect	[IPV_ADCDIG, ADC]Issue found by methodology configuration test for Calypso 4.2 RTM 100

Table continues on the next page...

ID	Subtype	Headline and Description
		'-- Converted from JIRA ticket MCAL-15104 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): -When implementing the methodology test for Calypso 4.2 RTM 100, I have found some issues: + Generate failure when enabling ConvTimeOnce and the node AdcNormalConvTimings is not exist (do not click on it), because of below condition: [!IF "((../../../NonAutosar/AdcEnableDualClockMode = 'true') or (../../../ NonAutosar/AdcConvTimeOnce = 'true') and (node:exists(AdcNormalConvTimings)))"!][!// /**< @brief Normal Conversion time - CTR0 */ (Adc_ConversionTimeType) (ADCDIG_INPSAMP_VALUE_U32([!"num:i(AdcNormalConvTimings/ AdcSamplingDurationNormal)"!])), [!ELSE!][!// (Adc_ConversionTimeType)0x0, [!ENDIF!][!/ And the default value 0x00 is incorrect. + Generate failure when enable the BCTU DMA transfer, config 2 units and only config DMA for BCTU_0 in MCL module, because of below conditons: [!FOR "Unit" = "0" TO "num:i($maxUnit)"!][!// [!VAR "mcl_Id"="0"!][!// [!VAR "foundDma"="0"!][!// [!VAR "UnitExits" = "0"!][!// [!LOOP "AdcConfigSet/BCTUHwUnit/*[1]/BCTU_InputTrigger/*/ AdcChannelTriggered/*"!][!// [!IF "ADCHWUNIT = concat('ADC',string(num:i($Unit)))"!][!// [!VAR "UnitExits"="$UnitExits+1"!][!// [!BREAK!][!// [!ENDIF!][!// [!ENDLOOP!][!// [!IF "AdcConfigSet/BCTUHwUnit/*[1]/BCTUDMAChannelEnable"!][!// [!LOOP "node:refs('ASPathDataOfSchema:/TS_T2D35M10I0R0/Mcl')/ MclConfigSet/*[1]/DMAChannel/*"!][!// [!IF "$UnitExits = 1"!][!// [!VAR "temp"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 16)))))"!][!// [!IF "$temp = concat('BCTU_',string(num:i($Unit)))"!][!// [!VAR "foundDma"="1"!][!// [!VAR "mcl_Id"="string(MclDMAChannelId)"!][!// [!ENDIF!][!// [!ENDIF!][!// [!ENDLOOP!][!// [!IF "$UnitExits = 1"!][!// [!IF "$foundDma = 0"!][!// [!ERROR "No DMA channels enabled and configured in the MCL Plugin for transfer conversion results for BCTU_ADC. Enable them into MCL Plugin."!][!// [!ENDIF!][!// + Build failure: When DMA used, MHT group enable, no Interrupt is enabled in Adc Interrupt. -Function Adc_Adcdig_EndMultipleCtuConv is built. -Function Adc_Adcdig_RecordResult is not built But in Adc_Adcdig_EndMultipleCtuConv we call Adc_Adcdig_RecordResult >> build failed. The error log:

Table continues on the next page...

ID	Subtype	Headline and Description
		STDERR:"C:/EB/EB20/tresos/plugins/Adc_TS_T2D35M10I0R0/src/ Adc_Adcdig_Irq.c", line 2116: error #223-D: STDERR: function Adc_Adcdig_RecordResult declared implicitly Test Case ID (internal TC that caught the bug) - optional Adc_TS_Eq_Cot_01",'N/A
ENGR00391400	NewWork	[LIN] Function define implicity warnings because of wrong including header structure in LinFlex software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT, LIN for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly error #223-D Function EXIT_INTERRUPT declared implicity *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00389470	Defect	[LIN] LinClockRef_Alternate shows error even when it is disable '-- Converted from JIRA ticket MCAL-14829 -- Problem detailed description (how to reproduce it): Enable LinClockRef_Alternate without value and then disable that parameter by setting LinEnableDualClockMode to FALSE produces an ERROR: The node "/AUTOSAR/TOP-LEVEL-PACKAGES/Lin/ELEMENTS/Lin/LinGlobalConfig/LinChannel/LinChannel_0/LinClockRef_Alternate" with value "" does not refer to nodes. Preconditions: Enable LinClockRef_Alternate without value and then disable that parameter by setting LinEnableDualClockMode to FALSE Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Error occurs even though LinClockRef_Alternate is disable Expected behavior: Error should be clean once LinClockRef_Alternate is disable 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): Please see the analysis tab.","'Modifile Lin.xdm, Lin_Cfg.h, Lin_Cfg.c, Lin_PBcfg.c
ENGR00391575	NewWork	[MCL] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure,ENGR00391576
ENGR00388571	Defect	[MCL] MclCrossbarHaltLowPriority doesn't supported from Raininer and wrong priority elevation 'Problem detailed description (how to reproduce it): Problem 1: From Rainier, Calypso Bit 1 on XBAR_CRSn is reserved bit but xdm still enable support this function. This function should enable for some derivative only for example Panther MPC574XP. Problem 2: From Rainier: Rainier doesn't support priority elevation for master (bit 8-15 on XBAR_CRSn). Therefore, it should disable for some derivative such as Rainier Problem 3: XBAR1 doesn't support Master Port No 0. However, CDD_Mcl_cfg_CrossbarPrioAndControl.m4 always get value from Port No 0 as default value to set value for XBAR_PRSn. In this case it write incorrect priority level. Preconditions: [...] Mcl_TS_008 [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]",'N/A
ENGR00388525	NewWork	[MCL] Support for running from User Mode 'NewWork Description: 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 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.

Table continues on the next page...

ID	Subtype	Headline and Description
		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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A
ENGR00388740	NewWork	[MCL] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.",'N/A

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00389926	Defect	[MCU] Build fail when MCU_CONFIGURE_CADDRN == STD_OFF '-- Converted from JIRA ticket MCAL-15143 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): [MCU] Build fail when MCU_CONFIGURE_CADDRN == STD_OFF Preconditions: MCU_CONFIGURE_CADDRN == STD_OFF Test Case ID (internal TC that caught the bug) - optional Any Trigger (not applicable in case of new features): MCU_CONFIGURE_CADDRN == STD_OFF Observed behavior (not applicable in case of new features): Build fail Expected behavior: Build pass 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): The commas at the end should be moved",'move commas at the end in the Mcu_aModeConfig
ENGR00390013	Defect	[MCU] DRET and FRET should be enabled (initialized) only on Power on Reset event or destructive reset reason 'Problem detailed description (how to reproduce it): The DRET and FRET is configured in Mcu_Init/Mcu_IPW_Init/ Mcu_MC_RGM_ResetInit function so the "Destructive/Functional" reset counter is reset each time Mcu_Init is called regardless of reset reason. Because of this, the "Destructive/Functional" Reset Escalation will not work as the counter is reset on every reset and never reach the threshold value. (if threshold value >1). Preconditions: - DRET and FRET are configured in the configuration time with threshold > 1. Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: - DRET and FRET are re-initialized on every reset, the "Destructive/Functional" Reset escalation never occurs. Expected behavior: - DRET are initialized only on Power on Reset event, the "Destructive" Reset escalation occurs when the number of "Destructive" reset reaches the threshold value(configured in DRET). - FRET are initialized only on Power on Reset event or destructive reset event, the "Functional" Reset escalation occurs when the number of "Functional" reset reaches the threshold value(configured in FRET). 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
		Check the reset reason, to - enable (re-configure) DRET to a value (different from 0) only on Power on Reset event - enable (re-configure) FRET to a value (different from 0) only on Power on Reset event and destructive reset event","'DRET and FRET should be enabled (initialized) only on Power on Reset event AUTOSAR_MCAL_MCU_UM.pdf
ENGR00390014	Defect	[MCU] DRET and FRET should be enabled (initialized) only on Power on Reset event or destructive reset reason 'Problem detailed description (how to reproduce it): The DRET and FRET is configured in Mcu_Init/Mcu_IPW_Init/ Mcu_MC_RGM_ResetInit function so the "Destructive/Functional" reset counter is reset each time Mcu_Init is called regardless of reset reason. Because of this, the "Destructive/Functional" Reset Escalation will not work as the counter is reset on every reset and never reach the threshold value. (if threshold value >1). Preconditions: - DRET and FRET are configured in the configuration time with threshold > 1. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: - DRET and FRET are re-initialized on every reset, the "Destructive/Functional" Reset escalation never occurs. Expected behavior: - DRET are initialized only on Power on Reset event, the "Destructive" Reset escalation occurs when the number of "Destructive" reset reaches the threshold value(configured in DRET). - FRET are initialized only on Power on Reset event or destructive reset event, the "Functional" Reset escalation occurs when the number of "Functional" reset reaches the threshold value(configured in FRET). 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 reset reason, to - enable (re-configure) DRET to a value (different from 0) only on Power on Reset event - enable (re-configure) FRET to a value (different from 0) only on Power on Reset event and destructive reset event","'DRET and FRET should be enabled (initialized) only on Power on Reset event Mcu_MC_RGM.c
ENGR00390954	Defect	[MCU] FXOSC or FIRC is chosen as system clock although these clock was disable '-- Converted from JIRA ticket MCAL-15007 -- 1) No error appear in EB when I chose System clock is FXOSC although this source clock was disable. And when I build code, the bit dedicate FXOSC auto enable ? This problem do not happen with case of PLL 2) reappear an Invalid Mode Configuration Interrupt as context below :

Table continues on the next page...

ID	Subtype	Headline and Description
		- Disable FXOSC clock - Chose FXOSC as system clock a Flag related to invalid mode Configuration must appear but its not.','correct x-path in Mcu.xdm
ENGR00391523	NewWork	[MCU] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.','improve including mcal header structure,ENGR00391526 ENGR00391528 ENGR00391530 ENGR00391732
ENGR00391615	NewWork	[MCU] Implement the change according to new Reference manual for Calypso 3M/6M RTM 1.0.4 ASR 4.0. 'NewWork Description: After review change according to new Reference manual for Calypso 3M/6M RTM 1.0.4 ASR 4.0. The report as below: Calypso 6M: MC_CGM - MC_CGM_AC3_SC - removed MC_ME - MC_ME_PCTL81 is deleted Calypso 3M: MC_CGM - MC_CGM_AC3_SC - removed Proposed solution (Optional): Update source code according above changes.','Review change according to new Reference manual for Calypso 3M/6M RTM 1.0.4 ASR 4.0.
ENGR00390818	Defect	[MCU] PRAMCx_PRCR1[P0_BO_DIS] and PRAMCx_PRCR1[P1_BO_DIS] configurable support 'Problem detailed description (how to reproduce it): Initial Description: The driver always clear PRAMCx_PRCR1[P0_BO_DIS] and

Table continues on the next page...

ID	Subtype	Headline and Description
		PRAMCx_PRCR1[P1_BO_DIS]/PRAMCx_PRCR1[P1_BO_DIS]. These bits should be configurable. Preconditions: NA Observed behavior: Driver not support to configure for PRAMCx_PRCR1[PRI],PRAMCx_PRCR1[P0_BO_DIS]/ PRAMCx_PRCR1[P1_BO_DIS]. So Ram controller always operates in below mode selected: Round Robin arbitration is selected. Port p1 read burst are optimized Port p0 read burst are optimized When can it be observed? (at configuration time, at runtime, at compile time?) at configuration time Expected behavior: User can configuration for PRAMCx_PRCR1[PRI],PRAMCx_PRCR1[P0_BO_DIS]/ PRAMCx_PRCR1[P1_BO_DIS] to select specify operation mode Reported release baseline: NA Proposal solution: Calypso 6M: PRAMC: PRAMC_0: Added PRI, P1_BO_DIS and P0_BO_DIS (must add in xdm, RegOperations) PRAMC_1: Added P0_BO_DIS PRAMC_2: Added P0_BO_DIS Calypso 3M: PRAMC: PRAMC_0: Added PRI, P1_BO_DIS and P0_BO_DIS (must add in xdm, RegOperations) PRAMC_1: Added P0_BO_DIS","'Calypso 6M: PRAMC: PRAMC_0: Added PRI, P1_BO_DIS and P0_BO_DIS (must add in xdm, RegOperations) PRAMC_1: Added P0_BO_DIS PRAMC_2: Added P0_BO_DIS Calypso 3M: PRAMC: PRAMC_0: Added PRI, P1_BO_DIS and P0_BO_DIS (must add in xdm, RegOperations) PRAMC_1: Added P0_BO_DIS",ENGR00391410
ENGR00389720	Defect	[MCU] Remove errata e10447 that was implemented on Calypso 4.0 RTM 1.0.2 '-- Converted from JIRA ticket MCAL-14720 -- Remove errata e10447 that was implemented on Calypso 4.0 RTM 1.0.2 Old implemented on Calypso RTM 1.0.2: " Note: Please ensure before entering STANDBY0 or LPU_RUN Mode, PMCDIG_MCR[LVD_IO_HI_REE] and PMCDIG_MCR[LVD_PD2_COLD_REE] bits are programmed to 0 (Eraata ERR010447).<p/> " Just description including in McuPowerMode in Mcu.xdm file ==> Remove errata that mean remove this description in xdm ==> no affect to functional code.",'Remove errata e10447 that was implemented on Calypso 4.0 RTM

Table continues on the next page...

ID	Subtype	Headline and Description
		1.0.2: Remove note in xdm file
ENGR00388832	NewWork	[MCU] Support for running from User Mode 'NewWork Description: 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 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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):","'Files modified: - Mcu.xdm - Mcu_Ipw.c",ENGR00388837 ENGR00388873 ENGR00388874 ENGR00391520
ENGR00392216	NewWork	[MCU] Update 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]",'Update code to support user mode
ENGR00390735	Defect	[PORT] Add description of unused port pins and PortNotAvailablePins to the user manual 'Problem detailed description (how to reproduce it): The driver supports to configure the unused port pins and PortNotAvailablePins under EB Tresos. I would be better to add such information in the user manual to tell the customers which pins will be configured and which MSCR will be configured by these container. e.g: MPC5746C 176 pin, PD[11]-PCR[59], PI[9]-PCR[137], P[10] -PCR[137] (with ENGR00390618, available PCR149~PCR263) and the available pins but not configured in PortContainer whill be configured in NotUsedPortPin container. These information will be very helpful for the customer to avoid some bad configuration which increase power consumption. 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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): NA","would be better to add such information in the user manual to tell the customers which pins will be configured and which MSCR will be configured by these container. e.g: MPC5746C 176 pin, PD[11]-PCR[59], PI[9]-PCR[137], P[10] -PCR[137] (with ENGR00390618, available PCR149~PCR263) and the available pins but not configured in PortContainer whill be configured in NotUsedPortPin container. These information will be very helpful for the customer to avoid some bad configuration which increase power consumption.
ENGR00389880	Defect	[PORT] Fix build error after Base is changed '-- Converted from JIRA ticket MCAL-15118 -- Detailed description (how to reproduce it, in case of bugs and new feature detailed description, in case of new features): After Base is changed, Port module appear the error when build test 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): NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fix this error","file change: Port_Siul2.c Port_Siul2_aGPIODirChangeability -> Port_Siul2_au16GPIODirChangeability
ENGR00389727	Defect	[PORT] Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2 '-- Converted from JIRA ticket MCAL-14876 -- The attachment 'AUTOSAR_MCAL_PORT_Checklist_for_Code_Review_Intermediate.xls' exceeded 50 characters filename CQ limit, it was therefore reduced. Fix the code after review against checklist for Calypso RTM 1.0.0 ASR 4.2','N/A,ENGR00391037
ENGR00390618	Defect	[PORT] The driver should support all Unused GPIO Pin Termination '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"

Table continues on the next page...

ID	Subtype	Headline and Description
		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: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA 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): NA","'(MSCR[149]~MSCR[263] for 176 pin device and MSCR[159]-MSCR[263] for 100 pin device) are configured by port driver. AN5220 chapter -"6.3 Unused GPIO Pin Termination"
ENGR00388735	NewWork	[PORT] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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. Requirement source: - ease of use. Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		- update UM to contain the required info.','N/A
ENGR00388869	Defect	[PWM] OPWMT interrupt should be disabled if notifications are not used 'Problem detailed description (how to reproduce it): For EMIOS channels confiugred in OPWMT mode, the interrupts are always enabled(FEN bit is set) and routed to DMA (DMA bit is set) . If notifications are not configured for the channel, there is no interrupt code to clear the FLAG bit after a trigger event occurs. This causes the EMIOS to send a flood of triggers. Preconditions: EMIOS channel in OPWMT mode, notifications disabled. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Flood of triggers setn by EMIOS (for example, to BCTU) Expected behavior: No flooding of triggers should occur. The trigger signals should happen only once per PWM period. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Disable the interrupts for OPWMT channels in case notifications are not used.','Pwm_eMios.c
ENGR00378869	NewWork	[PWM] Optimize SRAM usage for configured channels-Part2 'NewWork detailed description (how to reproduce it): The ticket ENGR00374183 improved SRAM consumption by using the number of configured channels to define the global variables for channels in the driver. However the following variables are not declared with PWM_GTM_ATOM_HW_CHANNELS_MAX. In Pwm_Gtm_Tom.c: static VAR(Pwm_Gtm_EdgeNotificationType, PWM_VAR) Pwm_GtmTom_aChannelNotifType[PWM_GTM_TOM_HW_CHANNELS_NO]; In Pwm_Gtm_Atom.c: static VAR(Pwm_Gtm_EdgeNotificationType, PWM_VAR) Pwm_GtmAtom_aChannelNotifType[PWM_GTM_ATOM_HW_CHANNELS_NO]; The customer request: To use PWM_GTM_ATOM_HW_CHANNELS_MAX to declare the above variables and update the related code accordingly. Preconditions: MPC5777M MCAL4.0 FBR1.1.1 CD5 Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The variable size is not declared as the number of configured channels. Expected behavior: The variable size is declared as the number of configured channels. To reduce the SRAM consumption.

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):",'N/A,ENGR00389973
ENGR00388528	NewWork	[PWM] Support for running from User Mode 'NewWork Description: 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 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));

Table continues on the next page...

ID	Subtype	Headline and Description
		#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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A
ENGR00390741	Defect	[PWM] The default period should be converted to integeral value 'Problem detailed description (how to reproduce it): The Pwm_PeriodType is uint32 or uint16 depend on the hardware module. In the configuration code, the default period is float as the corresponding parameter (PwmPeriodDefault) is float per Autosar specification. The value should be converted to integeral type during generation time to avoid compiler warning/error due to casting float/double to uint16/uint32 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The compiler error: floating-point value does not fit in required integral type Expected behavior: No compiler error: floating-point value does not fit in required integral type Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'N/A
ENGR00388732	NewWork	[PWM] User Manual for each driver should contain a "How To" configure chapter for advanced feateures 'NewWork Description: 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. 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);

Table continues on the next page...

ID	Subtype	Headline and Description
		- 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. Requirement source: - ease of use. Proposed solution (Optional): - update UM to contain the required info.",'N/A
ENGR00391442	NewWork	[SIUL2][PORT] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00389852	Defect	[SPI] Error on some TSB tag in Spi.xdm when use SPI unit does not support TSB '-- Converted from JIRA ticket MCAL-14976 -- Problem detailed description (how to reproduce it): Error message appear on some tags (TSBFrameSize, DsiCsIdentifier, SecondaryFrameSize, SecondaryDsiCsIdentifier) when use SPI unit(SPI_0 -> SPI_5) does not support TSB mode. Preconditions: Configuration use SPI_0 -> SPI_5 on EB Tresos. Test Case ID (internal TC that caught the defect) - optional NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: NA Observed behavior: Error message appear on some tags: TSBFrameSize, DsiCsIdentifier, SecondaryFrameSize, SecondaryDsiCsIdentifier when Configuration on EB tresos. 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): NA",'N/A
ENGR00388041	NewWork	[SPI] Fix the findings releated to Review checklist 'NewWork Description: Last line in a file shall be an empty line: List of files: Spi_VersionCheck_Inc.m Spi_VersionCheck_Src.m Spi_VersionCheck_Src_PB.m Requirement source: (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A","'Files affected: -Spi_VersionCheck_Inc.m -Spi_VersionCheck_Src.m -Spi_VersionCheck_Src_LT.m -Spi_VersionCheck_Src_PB.m
ENGR00391413	NewWork	[SPI] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro: Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (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
		Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00390725	NewWork	[SPI] Improve CC codebase base on remove compiler warnings task on jira '-- Converted from JIRA ticket MCAL-17193 -- These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. 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-initializers]@17"","'Files affected: -Spi_RegOperations.m",ENGR00391245
ENGR00388526	NewWork	[SPI] Support for running from User Mode 'NewWork Description: 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		<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. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'N/A
ENGR00390838	NewWork	[SPI] The array elements of struct type will build error with CodeWarrior compiler if not initialize size of array '-- Converted from JIRA ticket MCAL-17470 -- The Reported Baseline 'BLN_SMCAL_4.0_CARCASSONNE_NEXT_RELEASE' is not available in CQ, 'BLN_SMCAL_4.0_FOUNDATION_BASELINE' used instead. These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. Defect on jira but it should be NewWork on Clearquest, like improvement task.

Table continues on the next page...

ID	Subtype	Headline and Description
		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: ^ # 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,

Table continues on the next page...

ID	Subtype	Headline and Description
		# 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;","'Files affected: -Spi.h -Spi.c",ENGR00391247 ENGR00391268
ENGR00388024	Defect	 [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","'Files affected: Spi.xdm

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00389153	Defect	[SPI]The driver accesses the incorrect channel index pointer in slave mode '-- Converted from JIRA ticket MCAL-16302 -- These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): When the SPI device works in slave mode, the data is pushed in TXFIFO then automatically transmitted once the master triggers a new frame transmission. When all data is pushed into the TX FIFO, the driver moves to next channel by increasing the channel index pointer in function Spi_Dspi_IsrFifoRx. The driver is configured with following channels: - SpiChannelType = IB, SpiDataWidth = 8, SpiIbNBuffers = 128, SLAVE MODE - SpiChannelType = IB, SpiDataWidth = 16, SpiIbNBuffers = 2, MASTER MODE - SpiForceDataType = true When the driver receives last 5 frames from master, the driver has pushed all data into the TX FIFO, so the remaining data is 0, then the channel index pointer is still increased inside Spi_Dspi_IsrFifoRx. Because the only one channel is configured for slave job, the driver will use the incorect channel index pointer instead of the channel index pointer which was configured for slave job. Preconditions: The slave mode is used. Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: The driver uses the incorrect channel index pointer, this leads to protection hook is called in the customer application. In other words, the exeption has occured in the customer application. Expected behavior: The driver uses the correct channel index pointer. The driver does not cause protection hook or exceptions. 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: - In the Spi_Dspi_IsrFifoRx, because the driver supports only one channel in a job for SPI slave. There is no need to update the channel index pointer (pDspiDev->pcCurrentChannelIndexPointer++ when the remaining data is 0 (0u == pDspiDev->RemainingData). - In the slave mode, the driver only calls the function Spi_Dspi_JobTransferFifoFill to push data into the TX FIFO when the remaining data is not 0.","'Files affected: -Spi_DSPI.c
ENGR00391649	NewWork	[WDG] Function define implicity warnings because of wrong including header structure in Mcal software 'NewWork Description: Several warnings for implicit declaration of trusted functions are reported for e.g. MCL, MCU, PORT. for ex, on port driver: warning #223-D function Call_Port_Siul2_SetUserAccessAllowed_TRUSTED declared implicitly *_Trusted* call on Port core file Port_Suil2.c: MCAL.h with the help of the following macro:

Table continues on the next page...

ID	Subtype	Headline and Description
		Call_Port_Siul2_SetUserAccessAllowed_TRUSTED As users, we created Port_Trusted.h (a custom file in order to declare the prototype of this function): The main topic and request for support, is because it seems that there is no way that the MCAL modules (in this example PORT) provide a "user defined - custom" header where the declaration of trusted functions are located, this would help avoiding warnings like the reported. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Investigate on mcal including header file structure, focus into Mcu, Port, Mcl which affecting in customer raising.",'improve including mcal header structure
ENGR00389807	Defect	[WDG] Remove declaration of Wdg_Swt_SetMode from Wdg_Swt.h '-- Converted from JIRA ticket MCAL-15997 -- These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. Detailed description (how to reproduce it): Wdg_Swt.h declared Wdg_Swt_SetMode, but the function it is not defined in the WDG driver. The declaration of Wdg_Swt_SetMode from Wdg_Swt.h should be removed. MISRA will report it as a violation of undefined symbol. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Observed behavior: Wdg_Swt.h declared Wdg_Swt_SetMode, but the function it is not defined in the WDG driver. MISRA will report it as a violation of undefined symbol. Expected behavior: Wdg_Swt.h should not declare Wdg_Swt_SetMode. No MISRA violation reported. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The decalaration of Wdg_Swt_SetMode from Wdg_Swt.h should be removed.",'Wdg_Swt.h
ENGR00389830	Defect	[WDG] WdgExternalTriggerCounterRef description is confusing '-- Converted from JIRA ticket MCAL-15458 -- These ConfigAffected: S32GXX_4.2, IMXVS4_4.2 were removed as they are invalid in CQ currently. 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)

ID	Subtype	Headline and Description
		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...)","'WdgConfig.xml

4.3 RTM 1.0.3

ID	Subtype	Headline and Description
ENGR00387527	NewWork	[ADC] Adc_ChannelsLimitCheckingCfg array is accessed with incorrect index 'Problem detailed description (how to reproduce it): Adc_ChannelsLimitCheckingCfg is generated by the hardware logical id in ASR 4.2 and by hardware id in ASR 4.0. In Adc_Adcdig.c and Adc_Adcdig_Irq.c, the access to Adc_ChannelsLimitCheckingCfg is fixed by hardware id. So it can access to unknown memory. Preconditions: ADC limit checking feature is configured and used. ADC units are configured in the driver in such way that the logical ids do not match the hardware ids. Example: AdcUnit_0 logical id is mapped to hardware id ADC1. If AdcUnit_0 logical id is mapped to hardware id ADC0, the problem does not manifest. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: access to incorrect memory address Expected behavior: The driver shuld access the correct memory for configuration structures, even if the hw ids do not match the logical ids in the configuration. 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","'Files modified: Adc_Adcdig.c Adc_Adcdig_Irq.c
ENGR00382255	NewWork	[ADC] Adc_u32BctuDmaInternalBuffer is put to a unexpected section with compiler option -discard_zero_initializers 'NewWork Description: When user uses the compiler options -discard_zero_initializers, variables with intialization 0 shall be put to .bss section so Adc_u32BctuDmaInternalBuffer is put to .bss, not put to a section as expected User can use ADC driver with his compiler option, so ADC driver needs to be updated to be OK with different compiler options Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): #define ADC_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE #include "MemMap.h" VAR(uint32, ADC_VAR) Adc_u32BctuDmaInternalBuffer[ADC_MAX_HW_UNITS] [ADC_MAXIMUM_HW_CHANNELS]; #define ADC_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE #include "MemMap.h"","'Adc_Bctu.c Adc_Bctu.h
ENGR00381166	NewWork	[ADC] Add configuration option for optimizing conversions for HW triggered, ONESHOT groups (CPR-MCAL-799.adc) 'NewWork Description: Implement new requirement CPR-<TBC>.adc: "ADC driver shall provide a configuration option to enable speed optimizations for conversion processing of Adc Groups configured with ADC_CONV_MODE_ONESHOT and ADC_TRIGG_SRC_HW. When this parameter is enabled, no other type of Adc Group may be configured. Note: Other hardware specific restrictions may apply to the configuration." Add configuration option for optimizing conversions for HW triggered, ONESHOT groups. Add boolean parameter AdcOptimizeOneShotHwTriggerConversions in Adc.xdm to enable/disable the Adc driver optimization for HW Triggered groups, OneShot, Single access. If Enabled, other types of groups cannot be configured in ADC driver and the code for interrupt routine / Dma notification will be optimized for speed. Also, all groups will be restricted to have at most 8 channels configured. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add configuration option for optimizing conversions for HW triggered, ONESHOT groups. Add boolean parameter AdcOptimizeOneShotHwTriggerConversions in Adc.xdm to enable/disable the Adc driver optimization for HW Triggered groups, OneShot, Single access. If Enabled, other types of groups cannot be configured in ADC driver and the code for interrupt routine / Dma notification will be optimized for speed. Also, all groups will be restricted to have at most 8 channels configured.","'Add configuration option for optimizing conversions for HW triggered, ONESHOT groups",ENGR00384169
ENGR00387888	NewWork	[ADC] Add support for running on MPC5746C cut2.0,2.1 or MPC5748G cut2.0,3.0 for Calypso RTM release 'NewWork Description: Add support in driver for running on all four chip/CUTs combos. The checking of chip/CUT version shall be done at run-time Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The checking of the cut version shall be done in the <Driver>_Ipw.h where depending on the cut version a feature shall be enabled or not: #define <FEATURE_NAME>_AVAILABLE (< condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) The <FEATURE_NAME>_AVAILABLE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code the change shall be active only if <FEATURE_NAME>AVAILABLE is defined. #ifdef <FEATURE_NAME>_AVAILABLE if ((boolean)TRUE ==<FEATURE_NAME>_AVAILABLE) { code for the feature available } else { Code if the feature is not available } #else /*default behavior for other platforms where runtime decision is not needed*/ #endif Everything that needs decision at runtime must have a <FEATURE_NAME>_AVAILABLE macro associated. If the differences between CUTs are at plugin level (resource elements, configuration parameters), the plugin shall contain only the elements that are applicable on all CUTs. 2. The changes for errata management. The errata selection shall be done for current platform at runtime, based on cut version. The handling of a specific errata at runtime shall be defined in Driver_IPW.h using a definition as: #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE < condition for true using

Table continues on the next page...

ID	Subtype	Headline and Description
		RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) If the errata is available on both cuts, than the ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall not be defined at all The ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code, if the errata shall be handled at runtime the following code shall be followed: #ifdef ERR_<IP_NAME>_<ERR_NUMBER> #if (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE
		if (ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE == (boolean)TRUE) #endif /*ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE*/ { Add errata code } #endif /* (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) */ #endif /* (ERR_<IP_NAME>_<ERR_NUMBER>) */ Note: RUNS_ON_CUT is defined in Base in Soc_Ips.h",'add multiple cuts at runtime.
ENGR00386731	Defect	[ADC] Build failed in case threshold feature is enabled but no channel has any threshold configuration 'Problem detailed description (how to reproduce it): - AdcEnableThresholdConfiguration is enabled - There are no channels that have configured the threshold feature - Invoke Adc_ConfigureThreshold function - Build failed at this line: if(ADC_UNUSED_THRESHOLD_U8 != Adc_pCfgPtr->pChannels[Unit][Ch].u8ThReg) Because u8ThReg only exists under ADC_WDG_SUPPORTED preprocessor and ADC_WDG_SUPPORTED is defined when one of channels use threshold (AdcEnableThresholds enabled). Preconditions: N/A 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):","'Files modified: Adc.c Adc.h Adc_Ipw.h",ENGR00387545
ENGR00384501	Defect	[ADC] Correct the behavior of Adc_SetClockMode depending on AdcConvTimeOnce

Table continues on the next page...

ID	Subtype	Headline and Description
		'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.",'N/A,ENGR00386425
ENGR00387415	NewWork	[ADC] Create quality package for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Provide the following documents: Traceability: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix.xls Traceability warning report: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix_Warnings.xls VSMD report: AUTOSAR_MCAL_<MODULE>_VSMDReport.html Integration manual: AUTOSAR_MCAL_<MODULE>_IM.pdf User manual: AUTOSAR_MCAL_<MODULE>_UM.pdf DOORs DXL check result: AUTOSAR_MCAL_<MODULE>_CheckDriverReq.txt UML design: AUTOSAR_MCAL_<MODULE>_SDD.eap Exclusive area report: AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls CheckDefReport: AUTOSAR_MCAL_<MODULE>_CheckDefsReport.xlsx.

Table continues on the next page...

ID	Subtype	Headline and Description
		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",'
ENGR00385589	Defect	[ADC] Extern declarations of configured notification functions are generated only for the first config set 'Problem detailed description (how to reproduce it): In case Implementation Config Variant VariantPostBuild is used, the generated file Adc_PBcfg.c will not contain the extern declarations for configured notification functions for all config sets, but only for the first config set. This issue does not manifest in any functional way if the same notification functions are used across all configuration sets that are defined. It also means that if multiple configuration sets are used and each configuration set contains different notification function names, the ADC driver will not compile due to missing extern declaration of all notification functions. In Adc_PBcfg.c file, these macros are called with wrong parameter: [!CALL "Adc_Notification_Config", "Type"="'PC'"!] [!CALL "Adc_WdgNotification_Config", "Type"="'PC'"!] It should be: [!CALL "Adc_Notification_Config", "Type"="'PB'"!] [!CALL "Adc_WdgNotification_Config", "Type"="'PB'"!] Preconditions: Implementation Config Variant is configured as VariantPostBuild and more that one configuration set is configured Notification functions are configured for end of group conversions or watchdog events, and have different values in different configuration sets. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Extern declarations for notification functions form all config sets should be generated. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In Adc_PBcfg.c file, these macros are called with wrong parameter: [!CALL "Adc_Notification_Config", "Type"="'PC'"!] [!CALL "Adc_WdgNotification_Config", "Type"="'PC'"!] It should be: [!CALL "Adc_Notification_Config", "Type"="'PB'"!] [!CALL "Adc_WdgNotification_Config", "Type"="'PB'"!]",'Fixing the error relate to generate in PB configuration
ENGR00384707	NewWork	[ADC] Fix code review checklist findings 'NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		After review code review against checklist, Some violations code checklist were found. The report was attached in ticket ENGR00382115 [ADC] Perform code review against checklist for S32K14X RTM 1.0.0 Requirement source: Planned requirement (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Fix findings from the code review against checklist excel - Attach the new report after implement this ticket","'- Review code review against checklist. - The checklist template and coding guideline are enclosed in the attachment.
ENGR00387899	Defect	[ADC] HW triggered conversions with BCTU do not work - TRG_FLAG bit is stuck 'In Adc_EnableCTUTrigger API, BCTU trigger is configured before initializing the ADC hardware. If a pending trigger exists, it will cause the TRG_GLAG bit in BCTU_TRGCFG_n register to be stuck, and only a reset of the system will clear it. The order of initialization should be ADC, and then BCTU. In Adc_EnableHardwareTrigger, even if the order of initialization is correct (first ADC, then BCTU), there is still a timing issue that might cause TRG_FLAG bit to be stuck. The PWDN bit of ADC is cleared, and then after just a few instructions, the BCTU trigger is configured. There is not enough time, in normal execution, for the ADC hardware to fully get out of PWDN state. A while loop should be added to check that the ADC hardware has reached the desired power state whenever it is updated. Proposed solution (Optional): This was fixed for Calypso 4.2; please see the pull request http://sw-stash.freescale.net/projects/MCAL/repos/adc/pull-requests/63/overview Add a while loop checking the status of PWDN bit before configuring the BCTU. Also, make sure that the while loop is used in all the code that set or clear PWDN bit. Correct the order of hw initialziation in Adc_Bctu_EnableCtuTrigger() as below: if ((boolean)FALSE == bFirstTrigger) { #if (ADC_BCTU_AVAILABLE == STD_ON) Adc_Bctu_EnableCtuTrigger(Group, TriggerSource, (boolean)FALSE, bFirstTrigger); #endif /* ADC_BCTU_AVAILABLE == STD_ON */ /** @violates @ref Adc_Adcdig_c_REF_4 Return statement before end of function. */ return; } ..... #if (ADC_BCTU_AVAILABLE == STD_ON) Adc_Bctu_EnableCtuTrigger(Group, TriggerSource, (boolean)FALSE, bFirstTrigger); #endif /* ADC_BCTU_AVAILABLE == STD_ON */ /* Exit critical section */ SchM_Exit_Adc_ADC_EXCLUSIVE_AREA_21(); }','N/A

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00381300	NewWork	[ADC] Implement CPR-MCAL-791.adc (streaming groups performance) 'NewWork Description: Implement the new 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. " Requirement source: cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Implement CPR-<TBC>.adc,ENGR00386113
ENGR00385884	NewWork	[ADC] Implement requirement CPR-MCAL-792.adc for Adc_SetMode 'NewWork Description: Implement the new ADC requirement: CPR-MCAL-792.adc "ADC driver will stop the execution of Adc_SetMode API if there are any ongoing conversions at the moment of the call and retrun E_NOT_OK. In addition, if Development Error Tracing is enabled, ADC_E_BUSY will be reported." Requirement source: cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Change from: FUNC(Std_ReturnType, ADC_CODE) Adc_SetMode (VAR(Adc_SetModeType, AUTOMATIC) SetMode) { /* Return standard value */ VAR(Std_ReturnType, AUTOMATIC) TempReturn = (Std_ReturnType)E_NOT_OK; if ((Std_ReturnType)E_OK == Adc_ValidateSetMode(SetMode)) { TempReturn = Adc_Ipw_SetMode(SetMode); } return TempReturn; } to: FUNC(Std_ReturnType, ADC_CODE) Adc_SetMode (VAR(Adc_SetModeType, AUTOMATIC) SetMode) { /* Return standard value */

Table continues on the next page...

ID	Subtype	Headline and Description
		VAR(Std_ReturnType, AUTOMATIC) TempReturn = (Std_ReturnType)E_NOT_OK; if ((Std_ReturnType)E_OK == Adc_ValidateSetMode(SetMode)) { if ((Std_ReturnType)E_OK == Adc_ValidateCheckGroupNotConversion(ADC_SETMODE_ID)) { TempReturn = Adc_Ipw_SetMode(SetMode); } } return TempReturn; }",'TBC
ENGR00382665	NewWork	[ADC] Implement requirement for AdcInitialNotification - CPR-MCAL-796.adc 'NewWork Description: Implement CPR-MCAL-796.adc: The Adc driver shall provide an optional configuration parameter (named AdcEnableInitialNotification) for enabling an extra notification to be called for each Adc Group conversion, besides the ASR defined AdcNotification parameter. AdcEnableInitialNotification can only be enabled if AdcEnableSetChannel is also enabled. If AdcEnableInitialNotification is enabled, an extra notificataion function will become configurable in the Adc Group container. If configured, this notification will be called at the beginning of the interrupt routine, before updating andy HW registers or Group status. Note: This feature is intended to be used together with Adc_SetChannel service. The initial notification can be used by the user application to call Adc_SetChannel API before Adc driver updates the HW configuration for the next conversion. Requirement source: cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Support CPR-MCAL-796.adc,ENGR00384160
ENGR00382666	NewWork	[ADC] Implement requirement for Adc_SetChannel API - CPR-MCAL-797.adc 'NewWork Description: Implement CPR-MCAL-797.adc: The Adc driver shall provide an optional configuration parameter (named AdcEnableSetChannel) and an API for updating part of the group configuration at runtime. The optional API prototype shall be: FUNC(void, ADC_CODE) Adc_SetChannel(Adc_GroupType Group, Adc_GroupDefType * pChannel,uint16 * pDelays,uint16 u16Mask, Adc_ChannelIndexType NumberOfChannel); If configurable channel delays are not supported on the hardware platform, pDelays shall not be used. The service Adc_SetChannel shall reports error to DET (if enabled) when: - Adc_Init() was not called before this function - Parameter Group is out of range

Table continues on the next page...

ID	Subtype	Headline and Description
		- pChannel is NULL - pDelays is NULL (if configurable channel delays are supported by the hardware platform) - NumberOfChannel is 0 Adc_SetChannel() sevice shall update a secondary configuration for the Adc group that shall be used to reconfigure the associated HW registers. Adc driver shall check the secondary configuration from the interrupt routine functions (and Adc_ReadGroup for groups without interrupts) and update the HW registers in case it was updated. For the standard start APIs (Adc_StartGroupConversion, Adc_EnableHardwareTrigger) Adc driver shall also check if the secondary configuration was updated and use it, otherwise the primary group configuration will be used. Requirement source: cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Support Adc_SetChannel API,ENGR00384164 ENGR00387435 ENGR00387478
ENGR00377509	NewWork	[ADC] Improvement: Create macro file for ctu configuration 'NewWork Description: In RaceRunner Ultra 4.2, we have created a macro file for ctu template files. I can save time for developer if any CRs affect to template files in feature (update 1 file instead of 2). We should have what we did on RaceRunner Ultra for Panther, RR IS, Rainier and Calypso",',ENGR00381488
ENGR00383134	NewWork	[ADC] Internal refactoring of parameter validation checks 'NewWork Description: In Adc.c, Adc_SelfTest function : The argument for Adc_ValidateCallAndUnit function must be ADC_SELFTEST_ID not be ADC_CALIBRATE_ID One more things, in the Adc_Calibrate function, the validation for argument, pointer should be checked firsly. Adc_ValidateCheckGroupNotConversion and Adc_ValidateNotBusyNoQueue are so confusing because of the return. Adc_ValidateCheckGroupNotConversion should return E_OK when the group is not busy and E_NOT_OK when the group is busy. Adc_ValidateNotBusyNoQueue should return E_OK when unit is not busy and E_NOT_OK when the unit is busy. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): In Adc_SelfTest function, this below line of code should be changed: if ((Std_ReturnType)E_OK == Adc_ValidateCallAndUnit(ADC_CALIBRATE_ID, Unit)) to if ((Std_ReturnType)E_OK == Adc_ValidateCallAndUnit(ADC_SELFTEST_ID ,

Table continues on the next page...

ID	Subtype	Headline and Description
		Unit)) The validation for Pointer and argument should be called before check hardware is not under conversion.",'Fix Adc_SelfTest function reported wrong Service Id
ENGR00383170	NewWork	[ADC] Some Non_Autosar parameters should not be masked as optional 'NewWork Description: In the description of the ENGR00373357, it says: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). So, Non-Autosar parameters that have default value already should not be masked as optional. Non-Autosar containers should be masked as optional. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Remove all <a:a name="OPTIONAL" value="true"/> of Non-Autosar parameters that have default value already","'Remove all <a:a name="OPTIONAL" value="true"/> of Non-Autosar parameters that have default value already
ENGR00386581	NewWork	[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 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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): N/A","'Fix the defect that list of channels is not updated in the next starting time Files modified: Adc.c",ENGR00387718
ENGR00385739	NewWork	[ADC] The default value for node:fallback for NonAutosar features should be false 'NewWork Description: There are checks against some NonAutosar parameters that have the default value true: AdcEnableDualClockMode, AdcConvTimeOnce. By default, all NonAutosar features should be disabled, so the default value for node:fallback for NonAutosar features should be false. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Change the default value for node:fallback for NonAutosar features to false.','Change the default value for node:fallback for NonAutosar features to false.
ENGR00386579	NewWork	[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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): N/A","'Fix the defect that the setchannel feature can have wrong behavior when the variable is not initialized Files modified: Adc.c",ENGR00387719
ENGR00386578	Defect	 [ADC] error in compiling when both AdcEnableChDisableChApi and AdcEnableSetChannel are enabled simutaneously 'Problem detailed description (how to reproduce it): Error in compiling when both AdcEnableChDisableChApi and AdcEnableSetChannel are enabled simutaneously: "C:/EB/tresos/21.0.0/plugins/Adc_TS_T2D47M10I0R0/src/Adc.c", line 5639: error #20: identifier "pGroupPtr" is undefined Adc_IndividualGroupId = pGroupPtr->EnableChDisableChGroupIndex; Preconditions: AdcEnableChDisableChApi is enabled AdcEnableSetChannel is enabled N/A 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","'Fix the defect that Files modified: Adc.c
ENGR00385834	Defect	 [ALL/GENERAL] Xdm scheme contains duplicated attributes 'Problem detailed description (how to reproduce it): During reading of the MODULE.xdm files there are following warnings from Tresos studio: application Node /TS_T2D46M10I1R0/Mcu defines two attributes with the same name "UPPER-MULTIPLICITY" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Mcu defines two attributes with the same name "LOWER-MULTIPLICITY" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Mcu defines two attributes with the same name "DESC" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanGeneral/CanCounterRef defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanGeneral/CanMainFunctionWakeupPeriod defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes

Table continues on the next page...

ID	Subtype	Headline and Description
		application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanHardwareObject defines two attributes with the same name "DESC" which could not be merged. Removing one of the attributes application Node /AUTOSAR/TOP-LEVEL-PACKAGES/TS_T2D46M10I1R0/ELEMENTS/Can/CanConfigSet/CanConfigSet/CanHardwareObject/CanHardwareObject/CanTTHardwareObjectTrigger defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanHardwareObject/CanFilterMaskRef defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanHardwareObject/CanIdValue defines two attributes with the same name "LABEL" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanHardwareObject/CanFdPaddingValue defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanController/CanTTController defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanController/CanTTController defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanController/CanControllerBaudrateConfig/CanControllerCBT defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Can/CanConfigSet/CanController/CanWakeupSourceRef defines two attributes with the same name "EDITABLE" which could not be merged. Removing one of the attributes
		application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/VendorId defines two attributes with the same name "IMPLEMENTATIONCONFIGCLASS" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/VendorApiInfix defines two attributes with the same name "IMPLEMENTATIONCONFIGCLASS" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/SwPatchVersion defines two attributes with the same name "IMPLEMENTATIONCONFIGCLASS" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/SwMinorVersion defines two attributes with the same name "IMPLEMENTATIONCONFIGCLASS" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/SwMajorVersion defines two attributes with the same name "IMPLEMENTATIONCONFIGCLASS" which could not be merged. Removing one of the attributes application Node /TS_T2D46M10I1R0/Fee/CommonPublishedInformation/ModuleId defines two attributes with the same name "IMPL",',ENGR00387943

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00385210	NewWork	[ALL_GENERAL] Add support for running on MPC5746C cut2.0,2.1 or MPC5748G cut2.0,3.0 for Calypso RTM release 'NewWork Description: Add support in driver for running on all four chip/CUTs combos. The checking of chip/CUT version shall be done at run-time Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The checking of the cut version shall be done in the <Driver>_Ipw.h where depending on the cut version a feature shall be enabled or not: #define <FEATURE_NAME>_AVAILABLE (< condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) The <FEATURE_NAME>_AVAILABLE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code the change shall be active only if <FEATURE_NAME>AVAILABLE is defined. #ifdef <FEATURE_NAME>_AVAILABLE if ((boolean)TRUE ==<FEATURE_NAME>_AVAILABLE) { code for the feature available } else { Code if the feature is not available } #else /*default behavior for other platforms where runtime decision is not needed*/ #endif Everything that needs decision at runtime must have a <FEATURE_NAME>_AVAILABLE macro associated. If the differences between CUTs are at plugin level (resource elements, configuration parameters), the plugin shall contain only the elements that are applicable on all CUTs. 2. The changes for errata management. The errata selection shall be done for current platform at runtime, based on cut version. The handling of a specific errata at runtime shall be defined in Driver_IPW.h using a definition as: #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE < condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) If the errata is available on both cuts, than the ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall not be defined at all The ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code, if the errata shall be handled at runtime the following code shall be followed: #ifdef ERR_<IP_NAME>_<ERR_NUMBER> #if (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE
		if (ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE == (boolean)TRUE) #endif /*ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE*/ {

Table continues on the next page...

ID	Subtype	Headline and Description
		Add errata code } #endif /* (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) */ #endif /* (ERR_<IP_NAME>_<ERR_NUMBER>) */ Note: RUNS_ON_CUT is defined in Base in Soc_Ips.h",'
ENGR00387429	NewWork	[ALL_GENERAL] Create driver IS FMEA for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Create driver IS FMEA for Calypso 3M/6M RTM 1.0.3 ASR 4.0. 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",'
ENGR00387414	NewWork	[ALL_GENERAL] Create quality package for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Provide the following documents: Traceability: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix.xls Traceability warning report: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix_Warnings.xls VSMD report: AUTOSAR_MCAL_<MODULE>_VSMDReport.html Integration manual: AUTOSAR_MCAL_<MODULE>_IM.pdf User manual: AUTOSAR_MCAL_<MODULE>_UM.pdf DOORs DXL check result: AUTOSAR_MCAL_<MODULE>_CheckDriverReq.txt UML design: AUTOSAR_MCAL_<MODULE>_SDD.eap Exclusive area report: AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls CheckDefReport: AUTOSAR_MCAL_<MODULE>_CheckDefsReport.xlsx. 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",'
ENGR00387445	NewWork	[ALL_GENERAL] Perform UML design review against checklist for Calypso RTM 1.0.3 ASR 4.0 'NewWork Description: Review UML/design 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 UML design guideline verion 6.0, date Sep-2016: https://www.nxp.com/go/

Table continues on the next page...

ID	Subtype	Headline and Description
		235285152 UML design checklist verion 5.0, date Nov-2016: https://www.nxp.com/go/235285152 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The result of this activity will be the review checklist and this ticket is treated as "platform specific". Design Review checklists, both Intermediate and Final, should be added in GIT and attached to this ticket. New ticket will be raised for the UML design modifications resulted for the "Design Review" activity. This ticket will be used to update UML design, will be analyzed for all platforms and changes will be integrated for all affected platforms. For deviation from the Design Guideline cases, the ticket type shall be "Bug".','
ENGR00387434	NewWork	[ALL_GENERAL] Perform code review against checklist for Calypso RTM 1.0.3 ASR 4.0 'NewWork Description: Review code against checklist. Fill the review result to the checklist. he 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.",'
ENGR00387480	NewWork	[ALL_GENERAL] Perform requirement analysis for Calypso 3M/6M RTM 1.0.3 ASR 4.0 '(Communicate with Requirement Manager for latest cPRT, gMRD, EA blns. This ticket should be available at beginning of release at the latest) NewWork Description: Analyze requirements for Calypso 3M/6M RTM 1.0.3 ASR 4.0 The AUTOSAR_MCAL_<MDL>_CheckDriverReq.txt needs to be updated in reports folder Requirement source: CPRT baseline: 12.0 BLN_CPRT_SMCAL_4.0_01.12.00 GMRD baseline: 3.1 BLN_GMRD_SMCAL_4.0_03.02.00 EA baseline: 8.0 BLN_EA_SMCAL_4.0_01.07.00

Table continues on the next page...

ID	Subtype	Headline and Description
		SOW Baseline INTREQ_RSR_SMCAL_4.0_CALYPSO_RTM_1.0.3_I03 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'
ENGR00387512	NewWork	[ALL_GENERAL] Review change according to new errata document for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: - Review the list of erratum provided in "Log" sheet of each given reviewing template. - Fill the errata review template for your module with below rule: keeping only the erratum relating to your module and removing the others (not related to your module). Fill the information to column E, F, G, H, I in the template. Finally, attach the review result to this CR with this naming convention: SMCAL_Errata_review_<module>_C3M_cut2.0 or SMCAL_Errata_review_<module>_C3M_cut2.1 - The more errata information can be found in errata documents as mentioned in Requirement source. For Calypso 3M: target to support the latest cut 2.1 and 2.0 Some errata have been removed between the cuts and both cuts have had errata added. The added errata should be reviewed for affect in MCAL code. For Calypso 6M: target to support the latest cut 3.0 and 2.0 Calypso 6M - There are no new errata to be reviewed this time. All of them have been reviewed in the previous Calypso releases. There are only errata removed between the cuts at this time. Both 3M and 6M When moving from old cut to new cut, some of old erratum still exist in lastest cut (see in "Log" sheet of errata review template) but some others have been removed from latest cut (see in "Removed" sheet of errata review template). So, the modules which implemented workaround in code need to be reviewed again to see if we need to keep or remove the workaround. For Calypso 3M, the following modules need to be reviewed: PORT, DIO, MCL, MCU, SPI, GPT, FR, MCU, CAN and ADC. For Calypso 6M, the following modules need to be reviewed again: GPT, FR, MCU, CAN. Requirement source: Calypso 6M: Cut 2.0 MPC5748G_1N81M Mask Set Errata Rev .2, Jun2016 Calypso 3M cut 2.1: MPC5746C_Comparison_Summary_Ver0_0, 14-Sep-2016. Calypso 3M cut 1.1: MPC5746C_1N06M, Rev.4 July 2016 Calypso 6M cut 3.0: MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0, 31-Oct-2016. Calypso 3M cut 2.1: C3M_cut2.1_new_errata_20170113 (internal document), 13-Jan-17 (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>_CxM_cutx.x. If any

Table continues on the next page...

ID	Subtype	Headline and Description
		workaround is removed in the code this time, please take a note in "Remark" column to state that information. - 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".',''
ENGR00387937	NewWork	[Adc] Fix misra comment 'Fix misra comment,'
ENGR00382987	NewWork	[BASE] Add CSEC to MemMap files 'NewWork Description: Add CSEC to MemMap files Requirement source: gMRD Proposed solution (Optional): Update MamMap.h file and add CSEC specific sections",'Added CSEc CDD sections to MemMap.h
ENGR00387889	NewWork	[BASE] Add support for running on MPC5746C cut2.0,2.1 or MPC5748G cut2.0,3.0 for Calypso RTM release 'NewWork Description: Add support in driver for running on all four chip/CUTs combos. The checking of chip/CUT version shall be done at run-time Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The checking of the cut version shall be done in the <Driver>_Ipw.h where depending on the cut version a feature shall be enabled or not: #define <FEATURE_NAME>_AVAILABLE (< condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) The <FEATURE_NAME>_AVAILABLE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code the change shall be active only if <FEATURE_NAME>AVAILABLE is defined. #ifdef <FEATURE_NAME>_AVAILABLE if ((boolean)TRUE ==<FEATURE_NAME>_AVAILABLE) { code for the feature available } else {

Table continues on the next page...

ID	Subtype	Headline and Description
		Code if the feature is not available } #else /*default behavior for other platforms where runtime decision is not needed*/ #endif Everything that needs decision at runtime must have a <FEATURE_NAME>_AVAILABLE macro associated. If the differences between CUTs are at plugin level (resource elements, configuration parameters), the plugin shall contain only the elements that are applicable on all CUTs. 2. The changes for errata management. The errata selection shall be done for current platform at runtime, based on cut version. The handling of a specific errata at runtime shall be defined in Driver_IPW.h using a definition as: #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE < condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) If the errata is available on both cuts, than the ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall not be defined at all The ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code, if the errata shall be handled at runtime the following code shall be followed: #ifdef ERR_<IP_NAME>_<ERR_NUMBER> #if (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE
		if (ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE == (boolean)TRUE) #endif /*ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE*/ { Add errata code } #endif /* (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) */ #endif /* (ERR_<IP_NAME>_<ERR_NUMBER>) */ Note: RUNS_ON_CUT is defined in Base in Soc_Ips.h","'Files modified : - Soc_Ip.h
ENGR00373306	NewWork	[BASE] Update *_MemMap.h not to include Compiler.h and Mcal.h 'Problem detailed description (how to reproduce it): Issues appear when users modify the <Dvr>_MemMap.h and not include Compiler.h in it. Preconditions: User updates Mcl_MemMap.h and doesn't use the includes from our current stubs. Test Case ID (internal TC that caught the defect) - optional Int_TS_Co_001 Trigger: - Observed behavior: Compiler error. Expected behavior: No issue. 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): Include Mcal.h from files that require",'Updated eTimer files not to rely on Mcl_MemMap.h to include Mcal.h/Compiler.h
ENGR00375515	NewWork	[CAN] Add support for PREXCEN and EDFLTDIS in CAN_CTRL2 'NewWork Description: 2 new bits supported by hardware for CAN FD: PREXCEN and EDFLTDIS in CAN_CTRL2. They should be supported in tresos configuration. Note: If treerunner and/or RaceRunner Ultra already support these bits, need to port the feature from them. -> not implemented yet on TR and RRU Expected behavior: 2 new bits supported by hardware for CAN FD: PREXCEN and EDFLTDIS in CAN_CTRL2. They should be supported in tresos configuration. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Requirement source: NA Proposed solution (Optional): 2 new bits supported by hardware for CAN FD: PREXCEN and EDFLTDIS in CAN_CTRL2. They should be supported in tresos configuration.",'Add support for PREXCEN and EDFLTDIS in CAN_CTRL2,ENGR00381800
ENGR00387892	NewWork	[CAN] Add support for running on MPC5746C cut2.0,2.1 or MPC5748G cut2.0,3.0 for Calypso RTM release 'NewWork Description: Add support in driver for running on all four chip/CUTs combos. The checking of chip/CUT version shall be done at run-time Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The checking of the cut version shall be done in the <Driver>_Ipw.h where depending on the cut version a feature shall be enabled or not: #define <FEATURE_NAME>_AVAILABLE (< condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) The <FEATURE_NAME>_AVAILABLE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code the change shall be active only if <FEATURE_NAME>AVAILABLE is defined. #ifdef <FEATURE_NAME>_AVAILABLE if ((boolean)TRUE ==<FEATURE_NAME>_AVAILABLE) { code for the feature available } else { Code if the feature is not available } #else /*default behavior for other platforms where runtime decision is not needed*/ #endif

Table continues on the next page...

ID	Subtype	Headline and Description
		Everything that needs decision at runtime must have a <FEATURE_NAME>_AVAILABLE macro associated. If the differences between CUTs are at plugin level (resource elements, configuration parameters), the plugin shall contain only the elements that are applicable on all CUTs. 2. The changes for errata management. The errata selection shall be done for current platform at runtime, based on cut version. The handling of a specific errata at runtime shall be defined in Driver_IPW.h using a definition as: #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE < condition for true using RUNS_ON_CUT()> (boolean)TRUE: (boolean)FALSE) If the errata is available on both cuts, than the ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall not be defined at all The ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE shall be defined as TRUE or FALSE depending of its availability on the CUT. In the IP code, if the errata shall be handled at runtime the following code shall be followed: #ifdef ERR_<IP_NAME>_<ERR_NUMBER> #if (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) #ifdef ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE
		if (ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE == (boolean)TRUE) #endif /*ERR_<IP_NAME>_<ERR_NUMBER>_ACTIVE*/ { Add errata code } #endif /* (ERR_<IP_NAME>_<ERR_NUMBER> == STD_ON) */ #endif /* (ERR_<IP_NAME>_<ERR_NUMBER>) */ Note: RUNS_ON_CUT is defined in Base in Soc_Ips.h",'update to support runtime multiple cuts
ENGR00383124	Defect	[CAN] Can_FlexCan_UpdateMB_NoFD shall not contain padding update 'Problem detailed description (how to reproduce it): In Can_FlexCan_UpdateMB_NoFD function, the driver wrongly updates other message buffers with padding value when: -multiple controllers are used -at least 1 controller does not use FD and 1 controller uses FD mode. -Payload length is configured >8 Bytes In this case, for controller which does not use FD, the padding may overwrite the buffer descriptor of next MBs or write to memory over range. This might cause unpredictable behavior on drivers. Preconditions: -multiple controllers are used -have at least 1 controller not using FD and 1 controller using FD mode. -Payload length is configured >8 Bytes Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Could transmit data with controller which does not enable FD. Expected behavior: All controllers work properly in mentioned configuration (>=2 controller, at least

Table continues on the next page...

ID	Subtype	Headline and Description
		1 enable FD, at least 1 disable FD, FD payload >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): Remove following code in Can_FlexCan_UpdateMB_NoFD #if (CAN_FD_MODE_ENABLE == STD_ON) for (u8MbDataByte = PduInfo->length; u8MbDataByte < CAN_REAL_PAYLOAD_U8; u8MbDataByte++) { /* @violates @ref Can_Flexcan_c_REF_6 Violates MISRA 2004 Required Rule 17.4,pointer arithmetic other than array indexing used */ /* @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_WRITE8(FLEXCAN_MB_DATA_NO_FD((u8HwOffset), (uint32)(u8MbIndex), (uint32)(u8MbDataByte)), (uint8)(Can_pCurrentConfig->MBConfigContainer.pMessageBufferConfigsPtr[u8MbIndex].u8FdPaddingValue)); } #endif",'Removed padding update in Can_FlexCan_UpdateMB_NoFD
ENGR00385059	Defect	[CAN] E_NOT_OK is not handled in Can_FlexCan_ChangeBaudrate 'Problem detailed description (how to reproduce it): Can_FlexCan_ResetController returns E_NOT_OK due to a timeout(LPMACK not set). To execute the code inside Can_FlexCan_ChangeBaudrate() the return value from Can_FlexCan_ResetController it is checked to be OK in order to execute change baudrate. In Can_FlexCan_ChangeBaudrate() there is no "else" statement to handle E_NOT_OK case, returned from Can_FlexCan_ResetController. In this case change baud rate will not have any effect and the user will not be notified. 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): NA",'Update a case of CAN_NOT_OK in Can_FlexCan_ChangeBaudrate().,ENGR00386114 ENGR00386474
ENGR00383433	NewWork	[CAN] Fix compiler warning 'NewWork Description: In case user defines CAN_DEV_ERROR_DETECT as STD_OFF, some parameters (pCanControlerDescriptor, u8BaudrateIndex, u8CtrlId) ,in Can_Write function in Can.c file, was defined but never referenced. It leads to

Table continues on the next page...

ID	Subtype	Headline and Description
		compiler warnings. This CR will fix these compiler warnings. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update behavior of driver to prevent the warnings .','Fix compiler warning
ENGR00383120	NewWork	[CAN] Fix compiler warnings 'NewWork Description: Here are some compiler warning in the driver code: line 948: parameter "u8BaudrateIndex" was never referenced line 1410: parameter "Controller" was never referenced Find all the warnings that appear in the driver and fix them (if possible) Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA','Fix some compiler warning in the driver code
ENGR00382940	NewWork	[CAN] Improve generation for CanControllerCBT and CanControllerFdBaudrateConfig container 'NewWork Description: When the CanControllerCBT is disabled and user imports or configures a configuration which have invalid values, the EB tresos tool will run generator appear some errors. And when the CanControllerCBT is not supported, the code generator still checks and generate content of CanControllerCBT container. These issues also happen with CanControllerFdBaudrateConfig. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Modify xdm configuration and code generator relate to CanControllerCBT and CanControllerFdBaudrateConfig in order to fix above issues.','Improve XDM configuration and code generator for CanControllerCBT container
ENGR00387459	NewWork	[CAN] Improve implementation of separating message buffer size 'NewWork 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):","'Files modified: Can_Flexcan.h Reg_eSys_FlexCan.h Can_Flexcan.c",ENGR00387462
ENGR00387564	Defect	[CAN] Incorrectly determine the controller ID of MBs in multi transmission mode 'Problem detailed description (how to reproduce it): 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 Preconditions: N/A 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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): + 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",'Incorrectly determine the controller ID of MBs in multi transmission mode
ENGR00384466	NewWork	[CAN] Message Buffer Data Size should be configured seperately for each RAM block 'NewWork Description: Currently, the CAN driver has only one configuration parameter "Message Buffer Data Size" to configure the RAM message buffer data size for all RAM blocks. Because the maximum message buffer is depent on the message buffer data size, all RAM block use the same parameter, it would limit the number of message buffer. e.g: "Message buffer data size" = CAN_64_BYTES_PAYLOAD The CAN_FDCTRL[MBDSR0] = CAN_FDCTRL[MBDSR1] = CAN_FDCTRL[MBDSR2] = 3, so maximum Message buffer is 7x3 = 21. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Message Buffer Data Size should be configured seperately for each RAM block,ENGR00386928
ENGR00387567	Defect	[CAN] Rx FIFO Overflow is not processed 'Problem detailed description (how to reproduce it): 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 Preconditions: N/A 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		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",'N/A
ENGR00385680	NewWork	[CAN] Solve Review against Code Checklist findings 'NewWork Description: Update the code in order to solve warnings from code checklist Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the code in order to solve warnings from code checklist engr00384938",'N/A,ENGR00386545
ENGR00383053	NewWork	[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.",'Visible Can Clock From Bus option when the platform does not support
ENGR00387525	Defect	[CAN] Timeout duration value for polling the FRZACK bit is not reloaded in Can_FlexCan_SetFDENBitErrata() 'Problem detailed description (how to reproduce it): According to errata e10368 procedure for setting the FDEN bit, at step 11 it is need to wait the FRZACK bit of the CAN_MCR register to be set by the hardware or timeout is reached.

Table continues on the next page...

ID	Subtype	Headline and Description
		In the current driver already covered this issues, however there is incorrect in the timeout duration value at step 11 in Can_FlexCan_SetFDENBitErrata() function which resolve the errata e10368 , u32TimeoutCount is not reloaded CAN_TIMEOUT_DURATION before perform to wait u32TimeoutCount decreasing back to zero. This problem leads to incorrectly configure Can bit time. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: reload CAN_TIMEOUT_DURATION before perform to wait u32TimeoutCount decreasing back to zero 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",'missing to reset timeout value
ENGR00387084	NewWork	[CAN] Update driver according to ASR4.2.2 specific on RR Ultra RTM release 'NewWork Description: Update the driver according to the requirements on ASR 4.2.2. The updated requirements have been identified using the ENGR00384031. 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): [...]",'Update the driver according to the requirements on ASR 4.2.2
ENGR00385179	Defect	[CAN] Wrongly calculate the address of the message buffer 24th when CAN FD is enabled with CAN_32_BYTES_PAYLOAD 'The message buffers from the 3rd RAM block will be used (24~35 MBs) when payload it is 32 bytes. If we configure the 24th MB, the values corresponding to 12 MB from 2nd MB RAM block will be rewritten instead of writing the data to corresponding memory for 24 MB due to an address which is calculated incorrectly. Problem detailed description (how to reproduce it): My customer is evaluating CAN-FD through our MCAL 1.0.2. When CAN-FD is enabled, the FlexCAN RAM can be partitioned in blocks of 512 bytes. (Please refer to the Table 43-9 on MPC5748G RM Rev4)

Table continues on the next page...

ID	Subtype	Headline and Description
		When I configure all MB as 32bytes payload, Number of MBs can be set like below. RAM block 0 : 0~11 MBs RAM block 1 : 12~23 MBs RAM block 2 : 24~35 MBs But when we tried to select the 24th MB through MCAL, 12th MB was selected and it seems that below definition has error. #define FLEXCAN_MB_REGION(mb) \ ((((uint32)(mb)*((uint32)CAN_REAL_PAYLOAD_U8+ (uint32)ARBITRATION_FIELD)) +FLEXCAN_MB_REMAINING_BYTES_REGION)/(uint32)REGION_LENGTH) CE's comment: When the CAN FD is configureda with 32 byte payload, the last 32byte in each block region is not used, it is accumulated from block 0 to block 1 but the macro- FLEXCAN_MB_REGION does not take this into the account. Redefine the macro as follwing could resolve the issue. #define FLEXCAN_MB_REGION(mb) \ ((((uint32)(mb)*((uint32)CAN_REAL_PAYLOAD_U8+ (uint32)ARBITRATION_FIELD)) +FLEXCAN_MB_REMAINING_BYTES_REGION * (mb/ ((uint32)REGION_LENGTH/((uint32)ARBITRATION_FIELD+ (uint32)CAN_REAL_PAYLOAD_U8))))/(uint32)REGION_LENGTH) Preconditions: - CAN FD is enabled - PAYLOAD is 32 bytes. Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The address of message buffer 24th is not correctly calculated. Expected behavior: The address of message buffer 24th is correctly calculated. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]",'Update calculating the address of the message buffer 24th when CAN FD is enabled with CAN_32_BYTES_PAYLOAD,ENGR00386480
ENGR00382201	NewWork	[DIO] Add new variable 'Bit_NO' in Tresos DioChannelGroup container to let the user choose number of continuous channels in the group 'NewWork Description: Right now, in order to configure a channelGroup, the user needs to manually enter: - the channel group mask - the channel group offset The mask field already contains information about the offset, so the user needs to make sure these 2 fields are in sync. In order to ease this step, a new field will be added: Bit_NO, which defines the number of continuous channels in the group. Based on the value of this field and the value of the channel group offset, the channel group mask will be automatically computed. Requirement source: Improvment (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): Please see descirption above",'Add new variable 'Bit_NO' in Tresos DioChannelGroup container to let the user choose number of continuous channels in the group
ENGR00385453	NewWork	[DIO] Allow the user to configure channel group with the pins which are only readable or only writable 'NewWork Description: Dio driver should allow the user to configure channel group with the pins which are only readable or only writable instead of the pins which are both readable and writable as the current implementation. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update INVALID field of DioPortMask node in xdm file",'Update INVALID field of DioPortMask node in xdm file.
ENGR00385433	NewWork	[DIO] Implement requirement CPR-MCAL-824 (add a new parameter named DioPortBitNumber in container ChannelGroup container) 'NewWork Description: Implement new requirement CPR-MCAL-824: "A new integer parameter called DioPortBitNumber shall be added in Dio Channel Group container to configure the number of continuous bits with value 1 in the group. The 'Mask' variable shall be updated to have a CALCULATE button. When pressed, the configuration tool shall automatically compute the mask based on the values of the 'DioPortBitNumber' and 'Offset' variables. By default, 'DioPortBitNumber' field shall have value 1." Requirement source: cPRT (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Implement new parameter as defined by the requirement. If Bit_NO parameter is already implemented, it has to be renamed to DioPortBitNumber.",'Rename Bit_NO parameter to DioPortBitNumber.
ENGR00381256	NewWork	[DIO] When ReversePortBits is checked, the plugin does not correctly verify the channel group mask against list of available pins from resource files 'Problem detailed description (how to reproduce it): Supposing we have a 8 bit port, partially implemented in Hw. Only bits 0,1,2 and 3 are available. We want to configure a channel group with all 4 available pins and want to write value 0x7. This means we need to configure a channel group with offset 0 and mask 0x0F. Everything works fine while ReversePortBits option is not checked When ReversePortBits option is set, the user should configure the channel group with offset 4 and mask 0xF0. But the current implementation of the driver

Table continues on the next page...

ID	Subtype	Headline and Description
		does not allow the user to do this. An error is generated at configuration time, saying that the mask is mapped over not available pins in the port, while actually the mask is correct (because it is reversed). Preconditions: ReversePortBits checkbox is checked Test Case ID (internal TC that caught the defect) - optional Test cases in TS_011 Trigger: N/A Observed behavior: If we use a partially implemented port, the channel group mask generated in configuration may map on the not implemented pins in the port Expected behavior: The channel group validation conditions should not allow the user to select a channel group mask that is not mapped over the list of available pins in the port 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 INVALID condition of channel group mask variable in Dio.xdm to consider the value of ReversePortBits checkbox. Update the txt resource files to contain list of available pins for read/write in reversed mode too.","'When ReversePortBits is checked, the plugin does not correctly verify the channel group mask against list of available pins from resource files
ENGR00352194	NewWork	[ETHIF] add support for Autosar standard 4.2.1 'NewWork Description: Add support in MCAL for Autosar 4.2.1 standard. First target is a code drop for Cobra55. We need to be able to generate releases for both ASR4.0.3 and ASR4.2.1 standards using the files in SASW vob. Expected behavior: Implement deltas to support the ASR 4.2.1 standard Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Review and implement changes in code to support Autosar 4.2.1 standard.",'Add Support ASR 4.2.1
ENGR00387432	NewWork	[ETH] Create quality package for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Provide the following documents: Traceability: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix.xls Traceability warning report: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix_Warnings.xls VSMD report: AUTOSAR_MCAL_<MODULE>_VSMDReport.html Integration manual: AUTOSAR_MCAL_<MODULE>_IM.pdf User manual: AUTOSAR_MCAL_<MODULE>_UM.pdf DOORs DXL check result: AUTOSAR_MCAL_<MODULE>_CheckDriverReq.txt UML design: AUTOSAR_MCAL_<MODULE>_SDD.eap Exclusive area report: AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls

Table continues on the next page...

ID	Subtype	Headline and Description
		CheckDefReport: AUTOSAR_MCAL_<MODULE>_CheckDefsReport.xlsx. 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",'
ENGR00385051	Defect	[ETH] Generated Tx buffer address is not correct for second controller 'Problem detailed description (how to reproduce it): In the latest code, the generated address which is used by second controller is match to the same address of the first controller. This might lead to confliction of memory address. In case both controller are in used, the transmision might not work properly. Preconditions: 2 controller are used in 1 configuration. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Tx buffer address for 2 controller are identical Expected behavior: Each controller needs to have separated memory. 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 following line: (VAR(uint32, AUTOMATIC))(&Eth_TxBuffers[0][0]), /**< Address of TX Buffer */ to: (VAR(uint32, AUTOMATIC))(&Eth_TxBuffers[[!"num:i($BufCountIndex)"!]][0]), /**< Address of TX Buffer */",'N/A
ENGR00385048	NewWork	[ETH] Update driver to use pointer types instead of uint32 / uint64 'NewWork Description: Constructs like the following one (any use of CPU_TYPE macro) should be removed from the driver: #if (CPU_TYPE == CPU_TYPE_64) CONST(uint64, FR_CONST)DataAddr; /* Address of the message buffer data field */ #else CONST(uint32, FR_CONST)DataAddr; /* Address of the message buffer data field */ #endif This does not cause any problem. However, this cause trouble in managing differences between 2 platform. Expected behavior: Using pointer to store pointer address Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf,

Table continues on the next page...

ID	Subtype	Headline and Description
		Errata.pdf...)",'Used pointer to store memory address instead of uint32
ENGR00380003	NewWork	[ETH] Verify all Misra errors and comments part 2 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA",'N/A,ENGR00385974 ENGR00386481
ENGR00384332	Defect	[FEE] Core exception occurs if foreign blocks are in descending order of block number in memory 'Problem detailed description (how to reproduce it): The issue occurs only when the feature FEE_SWAP_FOREIGN_BLOCKS_ENABLED is ON and foreign blocks are not in ascending block number order in the memory. Steps to reproduce it: 1.Configure FEE_SWAP_FOREIGN_BLOCKS_ENABLED to ON 2.Write application blocks in descending order and then switch to boot loader or write bootloader blocks and then switch to application. Bellow it's an example of how block headers should appear ordered in the memory: Cluster Header Foreign Block Number 6 Header Foreign Block Number 5 Header Foreign Block Number 4 Header Foreign Block Number 3 Header Foreign Block Number 2 Header Foreign Block Number 1 Header 3.Call Fee_Init Everything is ok if foreign blocks are in increasing order of block number in flash sector like this: Cluster Header Foreign Block Number 1 Header Foreign Block Number 2 Header Foreign Block Number 3 Header Foreign Block Number 4 Header Foreign Block Number 5 Header Foreign Block Number 6 Header Preconditions: FEE_SWAP_FOREIGN_BLOCKS_ENABLED is ON foreign blocks are in descending order of block number in flash sector Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Core exception occurred when parsing foreign blocks. The exception occurs

Table continues on the next page...

ID	Subtype	Headline and Description
		inside Fee_MainFunction during processing of module initialization scheduled by Fee_Init. Expected behavior: Core exception should not occur when parsing foreign blocks 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 of Fee_GetForeignBlockIndex.",'Hard fault if foreign blocks are in descending order of block number in memory.
ENGR00373866	NewWork	[FEE] Fee returns block INVALID instead of INCONSISTENT when block does not exist in the flash 'Below is report from customer. from what I read in one of the latest MCAL (S32R274_MCAL_0_9_0_BETA_ASR_REL_4_2_REV_0001_20160205) the Rfc is not implemented also since the status INCONSISTENT is passed to upper layer only in Fee_JobWriteData and Fee_JobEraseImmediateDone (as this is done in my Bolero 3M MCAL). Please note that this deviates the #Rfc58294 which has following proposed solution. Solution proposal for MEMIf: SWS_MemIf_00065: Description for MEMIF_BLOCK_INCONSISTENT: Old text: The requested block is inconsistent, it may contain corrupted data. New text: 1. The requested block is inconsistent, it may contain corrupted data. 2. Block is NOT found. For the NvM module, an additional requirement must be added, for the NvM_ReadPRAMBlock API, mentioning the following: "The job of the function NvM_ReadPRAMBlock shall load the default values according to processing of NvM_RestorePRAMBlockDefaults (also set the job result to NVM_REQ_RESTORED_FROM_ROM) if the read request passed to the underlying layer fails (MemIf reports MEMIF_JOB_FAILED or MEMIF_BLOCK_INCONSISTENT) and if the default values are available." In the FEE module: extend SWS_Fee_00023: >> [SWS_Fee_00023] The function Fee_MainFunction shall check the consistency of the logical block being read before notifying the caller. If an inconsistency of the read data is detected, the function >> Fee_MainFunction shall set the job result to MEMIF_BLOCK_INCONSISTENT and call the error notification routine of the upper layer if configured. << [SWS_Fee_00023] The function Fee_MainFunction shall check the consistency of the logical block being read before notifying the caller. If an inconsistency of the read data is detected or if the << requested block can't be found, the function Fee_MainFunction shall set the job result to MEMIF_BLOCK_INCONSISTENT and call the error notification routine of the upper layer if configured. In the EA module: extend SWS_Ea_00104: >> [SWS_Ea_00104] The function Ea_MainFunction shall check the consistency of the logical block being read before notifying the caller. If an inconsistency of the block is detected (see SWS_Ea_00046 >> and SWS_Ea_00047), the function Ea_MainFunction shall set the job result to MEMIF_BLOCK_INCONSISTENT and call the error notification routine of the upper layer if configured.

Table continues on the next page...

ID	Subtype	Headline and Description
		<< [SWS_Ea_00104] The function Ea_MainFunction shall check the consistency of the logical block being read before notifying the caller. If an inconsistency of the block is detected (see SWS_Ea_00046 << and SWS_Ea_00047) or if the requested block can't be found, the function Ea_MainFunction shall set the job result to MEMIF_BLOCK_INCONSISTENT and call the error notification routine of the upper << layer if configured. Observed behavior: Fee returns block INVALID instead of INCONSISTENT when block does not exist in the flash? Expected behavior: Fee returns block INCONSISTENT when block does not exist in the flash",'Fee returns block INVALID or INCONSISTENT when block does not exist in the flash (depending upon parameter).
ENGR00383024	Defect	[FEE] Incorrect memory mapping of Fee variable 'Problem detailed description (how to reproduce it): The variable Fee_ForeignBlockConfig is incorrectly placed in section SEC_VAR_INIT_16 because the variable is not initialized and the variable type is not uint16, but a structure. Preconditions: Variable Fee_ForeignBlockConfig is only used when configuration FEE_SWAP_FOREIGN_BLOCKS_ENABLED is ON. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The variable Fee_ForeignBlockConfig is incorrectly placed in section SEC_VAR_INIT_16 because the variable is not initialized and the variable type is not uint16, but a structure. This will result in a compiler error with the IAR compiler. Expected behavior: The variable Fee_ForeignBlockConfig should be initialized and placed in the SEC_VAR_INIT_UNSPECIFIED section. No compiler error with the IAR compiler. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The variable Fee_ForeignBlockConfig should be initialized and placed in the SEC_VAR_INIT_UNSPECIFIED section.","'Updated Fee.c: The variable Fee_ForeignBlockConfig was initialized and placed in the SEC_VAR_INIT_UNSPECIFIED section.
ENGR00382629	NewWork	[FLS] Improve cache operations used in drivers to invalidate/flush data buffers 'NewWork Description: Current cache workaround implementation, used in MCL, ETH and FLS drivers for the data buffers, invalidates/flushes the entire cache.

Table continues on the next page...

ID	Subtype	Headline and Description
		Use cache operations targeted only on the cache lines which store the used buffers Proposed solution (Optional): Add new cache APIs to be used in drivers(MCL, ETH, FLS) which shall take as parameter the address and length of the memory area to be invalidated/ flushed, and perform those cache operations only on those cache lines. Make a check on the length parameter, and decide whether it is faster to perform the operation on the entire cache, if the memory area spreads across too many cache lines. Use the cache line operations in drivers for supporting the cache workaround instead of the global cache operations currently used. NOTE: For ETH and FLS, the change, if needed, shall be implemented only after this CR is implemented in MCL. At this moment, the new APIs are not yet available.','Add cache line invalidation functions.
ENGR00386139	Defect	[FLS] Investigate and fix MISRA 10.3,12.6 for equivalent boolean expressions 'Problem detailed description (how to reproduce it): The construct: bWasEccEvent = (0UL != REG_BIT_GET32(FLASH_MCR_ADDR32,FLASH_MCR_EER_U32)); generates MISRA 10.3 - The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression. The reason for the error is that the result of a comparison operator is an equivalent boolean of type signed integer, and the boolean type is defined as unsigned char in the current implementation. if((u32McrValue & FLASH_MCR_ERS_U32) || (u32McrValue & FLASH_MCR_PGM_U32)) The operands of logical operators (&&, ||and !) should be effectively Boolean. Expressions that are effectively Boolean should not be used as operands to operators other than Preconditions: NA Test Case ID (internal TC that caught the defect) - optional MISRA check Trigger: NA Observed behavior: MISRA 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): Use an "IF/ELSE" or "(expression) ? (boolean)TRUE : (boolean)FALSE" construct. and Use if((0U != (u32McrValue & FLASH_MCR_ERS_U32)) || (0U != (u32McrValue & FLASH_MCR_PGM_U32)))","'Fix MISRA 10.3,12.6 for equivalent boolean expressions
ENGR00383840	NewWork	[FLS] READONLY attribute is evaluated by XPATH accessing resource file

Table continues on the next page...

ID	Subtype	Headline and Description
		should be replaced by EDITABLE attribute 'Problem detailed description (how to reproduce it): During generation of Fls via command line, the following error occurs: Invalid XPath-expression for Attribute "READONLY" of node "/AUTOSAR/TOP-LEVEL-PACKAGES/MyProject238_ecuc/ELEMENTS/Fls/NonAutosar/FlsEnableUserModeSupport": (35016) Cannot get a resource file for target PA and derivate MPC574XG. Maybe this is due to module restrictions.; Preconditions: - The command line is used to generate configuration source code from arxml file. - The Fls configuration data position is placed before the resource configuration data in arxml file Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The error message appears, the configuration code is not generated successfully. Expected behavior: No error message appears, the configuration code is 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): Probably, this is an issue of EB Tresos since the error disappears when the resource configuration data is moved before the Fls configuration data. However we need to clarify with EB. In the meanwhile, Because the feature is not supported on MPC574XG, there is no benifit for the customer from the parameter-FlsEnableUserModeSupport, it should be removed or the READONLY attribute of this parameter should be hard code as "true" instead of a xpath referecing to the resource data. EB has confirmed this is Tresos issue. The proposed solution is to replace READONLY by EDITABLE attribute.",'N/A
ENGR00381934	Defect	[FLS] User manual refers chapter which does not exist in IM 'Problem detailed description (how to reproduce it): User manual specifies "please see chapter 5.6 User Mode Support in IM", but chapter "5.6 User Mode Support in IM" does not exist in IM. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: Customer Observed behavior: User manual specifies "please see chapter 5.6 User Mode Support in IM", but chapter 5.6 refers to another feature. Expected behavior: The user mode description in the UM and IM must be correct for all platforms, regardless if MCAL has user mode support or not. If the platform does not have user mode support, then chapter 5.6 should specify "MCAL does not include user mode support for this platform". 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): The user mode description in the UM and IM must be correct for all platforms, regardless if MCAL has user mode support or not. If the platform does not have user mode support, then chapter 5.6 should specify "MCAL does not include user mode support for this platform".","'Add chapter 5.6 User Mode Support in IM. "MCAL does not include user mode support for this platform". since MPC574xG did not support this feature.
ENGR00379501	NewWork	[FLS] Verify all Misra errors and comments Part 2 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls After first phase, Bellow Misra violations need more time to analysis and will be fixed in the next phase if needed. MISRA 2004 Rule 17.4(Required) MISRA 2004 Rule 12.4(Required) MISRA 2004 Rule 14.7(Required) MISRA 2004 Rule 11.4(Advisory) MISRA 2004 Rule 8.12(Required) MISRA 2004 Rule 17.2(Required) MISRA 2004 Rule 10.3(Required) Fls_Cfg.h: "CONSTP2CONST(Fls_AddressType, FLS_CONST, FLS_APPL_CONST) pSectorEndAddr;" transformed to: "CONST(Fls_AddressType, FLS_CONST) (*pSectorEndAddr)[];" It should need an extra CONST, like: "CONST(Fls_AddressType, FLS_CONST) (* const pSectorEndAddr)[];" the same for all the others pointer to arrays changed. Fls_LLD_Code.m4: In Fls_Flash_Init(), line 355, the MISRA comment "fls_c_REF_10" is no longer needed, can be deleted. In Fls_Flash_MainFunction(), line 1409,1539,1864 for: (void) Fls_Flash_FinishHVOperation(Fls_Flash_pRegBasePtr); , a comment can be added to explain the cast and ignore of the return value. In Fls_Flash_SectorWrite(), line 3352, the MISRA comment "fls_c_REF_10" is no longer needed, can be deleted. At the end of Fls_Flash_SectorErase, a DEM error should be added if the Fls_Flash_VerifyErase function fails. Output expected: NA Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Fix all MISRA errors that are not in in MISRADeviation.xls,ENGR00385914 ENGR00386215 ENGR00386490

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00387412	Defect	[FLS]Wrong time out value for abort case 'Problem detailed description (how to reproduce it): Wrong timeout value for abort case Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: At run-time Observed behavior: pTimerCounterAC is used instead of pTimerCounterAbort Expected behavior: Replace pTimerCounterAC with pTimerCounterAbort Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Replace pTimerCounterAC with pTimerCounterAbort in the function Fls_Flash_AccessCode in the abort case","'Files modified: Fls_Ac.c
ENGR00366841	NewWork	[FR] Change Flexray Message Buffer sharing algorithm 'NewWork Classification: Customer requested to modify the algorithm used at configuration time in order to allow that 4 LPDUs to share the same physical message buffer. Current implementation of the configuration algorithm allows only 2 LPDUs to share the same physical message buffer if some conditions are also meet.","'Java Fr_CfgPb.c
ENGR00385945	NewWork	[FR] Create De-initialization function 'NewWork Description: CPR-MCAL-805.fr - The FR driver shall provide a de-initialization function. Requirement source: Internal requirement Proposed solution (Optional): Create the Fr_DeInit compatible with CPR-MCAL-805.fr","'Fr.xdm Fr.c Fr.h Fr_IPW.c Fr_IPW.h Fr_Cfg.h",ENGR00386544
ENGR00387433	NewWork	[FR] Create quality package for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Provide the following documents: Traceability: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix.xls Traceability warning report: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix_Warnings.xls VSMD report: AUTOSAR_MCAL_<MODULE>_VSMDReport.html

Table continues on the next page...

ID	Subtype	Headline and Description
		Integration manual: AUTOSAR_MCAL_<MODULE>_IM.pdf User manual: AUTOSAR_MCAL_<MODULE>_UM.pdf DOORs DXL check result: AUTOSAR_MCAL_<MODULE>_CheckDriverReq.txt UML design: AUTOSAR_MCAL_<MODULE>_SDD.eap Exclusive area report: AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls CheckDefReport: AUTOSAR_MCAL_<MODULE>_CheckDefsReport.xlsx. 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",'
ENGR00386462	NewWork	[FR] Update driver ASR403 to synchronize ASR422 'NewWork Description: Update the driver according to the requirements on ASR 4.2.2. The updated requirements have been identified using the ENGR00384031. 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...)",'All files in driver
ENGR00385049	NewWork	[FR] Update driver to use pointer types instead of uint32 / uint64 'Problem detailed description (how to reproduce it): Constructs like the following one (any use of CPU_TYPE macro) should be removed from the driver: #if (CPU_TYPE == CPU_TYPE_64) CONST(uint64, FR_CONST)DataAddr; /* Address of the message buffer data field */ #else CONST(uint32, FR_CONST)DataAddr; /* Address of the message buffer data field */ #endif Instead, the associated data types shoudl be changed from uint32 / uint64 to pointer type. Preconditions: N/A Observed behavior: N/A When can it be observed? (at configuration time, at runtime, at compile time?) N/A Expected behavior: Don't use the uint32 or uint64 type

Table continues on the next page...

ID	Subtype	Headline and Description
		Reported release baseline: N/A Proposed solution (Optional): Replate the uint32/uint64 by a pointer. NewWork Classification: (internal task, improvement, feature request) N/A In case for feature request, does a TWG ticket exist (cloned or linked)? Yes NewWork Description: N/A Expected behavior: N/A Requirement source: N/A","'Fr_CfgPb.c Fr.h Fr_Flexray.c",ENGR00386464
ENGR00381385	NewWork	[FR] Verify all Misra errors and comments (part 2) 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls In the part 1, some violation not yet fixed. That are violations for rule 17.4, 19.4, 1.2 and 11.4. In order to fix these violates, need to discuss carefully with teach lead to figure out the best solution. Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: All the misra exclude into deviation document which must fix successful Requirement source: Missra document 2004 Proposed solution: Analysis, discuss and fix misra error.','Update code in order to fix misra errors
ENGR00381957	Defect	[FR] Wrong allocation variable in memory map 'Problem detailed description (how to reproduce it): When declaring global variable in Fr.h, variable Fr_VirtualResourceAllocation is allocated in Non-initialized section, but in file Fr.c, it is defined in initialized section. This might lead to issue depending on compiler options. Note: in current environments, this issue does not cause any problems. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: Put Fr_VirtualResourceAllocation array to initial section. Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		Move variable Fr_VirtualResourceAllocation from "...SEC_VAR_NO_INIT_16" to "...SEC_VAR_INIT_16"","'Fr.c: correct pre-define "...SEC_VAR_NO_INIT_16" to "...SEC_VAR_INIT_16"
ENGR00385986	Defect	[FR]Fix misra error. 'Problem detailed description (how to reproduce it): have many misra errors in driver. In the Fr.c: violate rule 8.7; rule 19.1 Fr_Flexray.c: violate rule 11.4; rule 11.5; rule 19.1; rule 11.3 Fr_Cfg.h: violate rule 19.4 Fr.h: violate rule 8.12 Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: All misra is resloved Proposed solution (Optional): fix or comment all misra violation.","'Fr.c Fr_Flexray.c Fr.h Fr_Cfg.h
ENGR00383776	Defect	[FR]Fix misra violation. 'Problem detailed description: In the Fr driver: - Fr_Flexray.h file: code has some MISRA violations related to rule 5.1 - Fr_Ipw.h file: driver contains some violations related to rule 1.4 All these rules are included to deviation document (cannot be fixed). The solution for solving this issue was adding comments for them in the code. There is no functional impact to the driver. Preconditions: N/A Test Case ID: N/A Trigger: N/A Observed behavior: These MISRA violations only impact the automatic generation of MISRA report. They don't impact to functionality. Expected behavior: All MISRA errors shall be fixed or commented Proposed solution: add comments for them.','N/A,ENGR00383777
ENGR00387908	NewWork	[Fr] Fix misra comment

Table continues on the next page...

ID	Subtype	Headline and Description
		'Fix misra comment,'
ENGR00387759	Defect	[GPT] Fix misra errors 'Problem detailed description (how to reproduce it): Fix misra 8.7 error that appears in a certain configuration for Gpt_eMode. Preconditions: Gpt_eMode only appears used from one function when only Gpt_Init API is used, raising misra 8.7: Objects shall be defined at block scope if they are accessed from withing a single function. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Misra error 8.7. Expected behavior: No misra error. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Comment misra 8.7.",'Gpt.c,ENGR00387776
ENGR00381715	NewWork	[GPT] 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 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*,

Table continues on the next page...

ID	Subtype	Headline and Description
		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: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Files Modified: Gpt_Gtm.c
ENGR00382214	Defect	[GPT] Possible delay when updating PIT_RTI_LDVAL register 'Problem detailed description (how to reproduce it): PITRTI might have issues when updating PIT_RTI_LDVAL. as shown in the RM, in the chapter for the RTI_LDVAL register. From the user's point of view: For some frequencies, PIT_RTI_LDVAL can have a delay in being programmed, causing the timer to function with the old load value for a number of clock cycles, after Gpt_StartTimer is called. In this time, CVAL can be loaded with the old value and cause the first period to be wrong. Preconditions: PIT_RTI is used. Test Case ID (internal TC that caught the defect) - optional Gpt_TC_0093 Trigger: at run-time Observed behavior: According to the RM In the case of the RTI, it will take several cycles until this value is synchronized into the RTI clock domain. For all other timers the value change is visible immediately. The synchronisation mechanism allows 0 wait states in this case.

Table continues on the next page...

ID	Subtype	Headline and Description
		PIT_RTI_LDVAL is not loaded so PIT_RTI_CVAL is not updated as expectation Expected behavior: Need to wait until finish PIT_RTI_LDVAL update 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 define a time-out parameter - Whenever PIT_RTI_LDVAL is updated, it will wait for that time-out to expire before PIT-RTI is enabled.','Gpt_pit.c,ENGR00387162
ENGR00382532	NewWork	[GPT] Register operations should not be redefined by macros 'NewWork Description: Register operations should not be redefined by macros EG: #define PIT_READ_TIMERVALUE_U32(u8ModuleIdx,u8ChannelIdx) (REG_READ32(PIT_CVAL_LOCKABLE_ADDR32((u8ModuleIdx), (u8ChannelIdx)))) This leads to a harder understanding of the code by the developer and the customer and may mask some other issues. Requirement source: Internal refactoring. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Replace IP level macros with REG operations from SilRegMacros.h','Gpt_Pit.c,ENGR00382534 ENGR00382535
ENGR00387159	Defect	[GPT] 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. in Gpt_PluginMacros.m, replace line 281: (EcuM_WakeupSourceType)[!IF "node:exists(GptWakeupConfiguration)"!] EcuMConf_EcuMWakeupSource_[!"as:ref(GptWakeupConfiguration/GptWakeupSourceRef)/@name"!][!ELSE!]0U[!ENDIF!], /* Wakeup information */

Table continues on the next page...

ID	Subtype	Headline and Description
		by: (EcuM_WakeupSourceType)[!IF "node:exists(GptWakeupConfiguration)"!] ((uint32)((uint32)1<<[!"as:ref(GptWakeupConfiguration/GptWakeupSourceRef)/ EcuMWakeupSourceId"!]))[!ELSE!]0U[!ENDIF!], /* Wakeup information */ 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):","'File modified: Gpt_PluginMacros.m Updating at line 500, Replace of */!]EcuMConf_EcuMWakeupSource_[!"as:ref(GptWakeupConfiguration/ GptWakeupSourceRef)/@name"!][!/*
		by */!]1<<[!"as:ref(GptWakeupConfiguration/GptWakeupSourceRef)/ EcuMWakeupSourceId"!][!/*
ENGR00383031	Defect	[ICU] Fix errors when optional references are not disabled in the plugin. 'Problem detailed description (how to reproduce it): Nodes that were changed from ENABLE to EDITABLE (IcueMiosChannelRef, IcuSiul2ChannelRef, IcuWkpuChannelRef) and have an optional field generate the error "Duplicate physical channel" when they are not disabled before the IcuHwIP value is changed. Also when configurations from previous releases (before the change to EDITABLE for some nodes) is imported as .xdm. Preconditions: User imports ussing .xdm files. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Issues when having multiple configurations with no reference to a Wkpu / Siul2 channel. 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 source code at .xdm and .m files.","'File modified: Icu_42.xdm Adding empty checking condition for reference nodes. <a:tst expr="not(text:uniq(../../*/IcueMiosChannelRef, .)) and (string(node:value(.)) != '')" true="Duplicate physical channel"/>

Table continues on the next page...

ID	Subtype	Headline and Description
		Also,please check in .m file at if existing [!IF "node:exists(IcueMiosChannelRef)"!] command line then replace by [!IF "contains(IcuHwIP,'EMIOS')"!] and same goes for other references node which contain optional attribute.
ENGR00385812	NewWork	[ICU] Removing the unnecessary exclusive area for Siul2 IPV (exclusive areas) 'NewWork Description: eview 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 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
		Requirement source: - Customer Request - AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls - Exclusive areas defined in driver code (all of *.c *.h related files). - Updated IM chapter 5.1 according to exclusive areas defined in the code

Table continues on the next page...

ID	Subtype	Headline and Description
		(IM.pdf) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Removing the unnecessary exclusive area for Siul2: ICU_EXCLUSIVE_AREA_29(); ICU_EXCLUSIVE_AREA_30();",'Review the implementation of data consistency mechanism (exclusive areas),ENGR00386430
ENGR00385082	Defect	[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):","'File modified: Icu_PluginMacros.m Updating at line 500, Replace of */!]EcuMConf_EcuMWakeupSource_[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/@name"!][!/* by

Table continues on the next page...

ID	Subtype	Headline and Description
		/!]1<<[!"as:ref(IcuWakeup/IcuChannelWakeupInfo)/EcuMWakeupSourceId"!][!/
ENGR00382663	NewWork	[LIN] Check the consistency of all defines 'NewWork Description: Generate the report for consistency of all defines by running CheckDefs tool and create <Driver>_MCAL_CheckDefsReport.xlsx Resolve all issues reported and for all false positives add a comment in the <Driver>_MCAL_CheckDefsReport.xlsx Expected behavior: File shall be <Driver>_MCAL_CheckDefsReport.xlsx and shall be attached to the CR. Requirement source: Customer Request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Run ChkDefs.pl in order to obtain the csv report as below perl ChkDefs.pl -o <Driver>_MCAL_CheckDefsReport.csv -r -f ../../output/plugins/<DriverPluginName>/*.c -f ../../output/plugins/<DriverPluginName>/*.h -a -s "^(\w+_\w+_)\w+~$1\w+"","'Generate the report and fix the warnings: - Reg_eSys_LINFlex.h
ENGR00387160	Defect	[LIN] 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 For Lin driver, in Lin_Cfg.c and Lin_PBcfg.c, replace line : EcuM_WakeupSourceType)EcuMConf_EcuMWakeupSource_[!"as:ref(LinChannelEcuMWakeupSource)/@name"!] by: (EcuM_WakeupSourceType)(1<<[!"as:ref(LinChannelEcuMWakeupSource)/EcuMWakeupSourceId"!]) Preconditions: The EcuM defines the EcuMWakeupSource as thevalue configured for

Table continues on the next page...

ID	Subtype	Headline and Description
		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):","''File modified: Lin_Cfg.c Lin_PBcfg.c Replace of (EcuM_WakeupSourceType)EcuMConf_EcuMWakeupSource_[!"as:ref(LinChannelEcuMWakeupSource)/@name"!] with
		(EcuM_WakeupSourceType)(1<<[!"as:ref(LinChannelEcuMWakeupSource)/EcuMWakeupSourceId"!])
ENGR00387551	Defect	[LIN] Update status for LIN chanel when slave send wake up signal 'Problem detailed description (how to reproduce it): When Master node in sleep mode, slave node sent a wake up signal to Master. However, status of master LIN still not change (LIN_CH_SLEEP_STATE) while wakeup signal transfer success. Preconditions: - Master in sleep mode. - MAF sent a wakeup signal Test Case ID (internal TC that caught the defect) - optional tc_fnc_lin_00206 Trigger: NA Observed behavior: Master did not change status channel, instead is LIN_CH_OPERATIONAL. Expected behavior: Master change to true status. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In function Lin_Linflex_TxRxInterruptHandler, after driver dectected a wakeup signal (WUF was set) but did not update some status variables: Lin_au8LinChStatus, Lin_au8LinChFrameStatus. * Detail: - Modified file: Lin_LINFlex.c => function Lin_Linflex_TxRxInterruptHandler - Add a block: /* Set Channel state Operational */ Lin_au8LinChStatus[u8LogicalChannel] = LIN_CH_OPERATIONAL; /* Update LIN channel frame operation status to LIN_CH_READY_STATE */ Lin_au8LinChFrameStatus[u8LogicalChannel] = LIN_CH_READY_STATE; after call fuction: EcuM_CheckWakeup()","''Update status for LIN chanel when slave send wake up signal

Table continues on the next page...

ID	Subtype	Headline and Description
		Files modified: Lin_LINFlex.c
ENGR00373951	NewWork	 [LIN] Use "EDITABLE" instead of "READONLY" in Lin.xdm 'NewWork Description: The 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/> Nodes that have READONLY attributes with type xPath will should change to use EDITABLE",'Lin.xdm
ENGR00385817	Defect	 [LIN] remove the excess commas in Lin_LINFLEX.c 'Problem detailed description (how to reproduce it): The array Linflex_au32BaseAddrs end with commas, CONST(uint32, LIN_CONST) Linflex_au32BaseAddrs[] = { #ifdef LINFLEX0_BASEADDR LINFLEX0_BASEADDR, #else ((uint32)0x00000000UL), #endif /* LINFLEX0_BASEADDR */ #ifdef LINFLEX1_BASEADDR LINFLEX1_BASEADDR, #else ((uint32)0x00000000UL), #endif /* LINFLEX1_BASEADDR */ .... #ifdef LINFLEX17_BASEADDR LINFLEX17_BASEADDR, #else ((uint32)0x00000000UL), #endif /* LINFLEX17_BASEADDR */ }; End this make a issue "too many initializer values" when build the source code","'File modified : + Lin_LINFlex.c
ENGR00381210	NewWork	 [MCL] Code improvements to conform with coding style guide 'NewWork Description: 1.Mcl_Axbs.c Don's use two levels of macros(AXBS_PRS_CONFIG, AXBS_CRS_CONFIG), use directly the REG_WRITE for writing into registers. 2.Delete the #ifdef guards for the main MCL features eg: #ifdef MCL_ENABLE_DMA,#ifdef MCL_ENABLE_AXBS, etc.. Since the main features are included by M4 preprocessing the #ifdefs ar enot needed anymore. Requirement source: NA (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): Code improvements to conform with coding style guide","'- reduce the nesting of macro definition - remove unessary guarding macro of features",ENGR00386438
ENGR00377785	NewWork	[MCL] Improve the APIs which set and get CITER and BITER 'NewWork Description: Cuurently the APIs have un-uniform approach(16 bit or 32 bit access, rmw or reg_Write). Improve the code so that the functions -use similar approach -don't directly access underlaying hardware, call macros from IPV_DMA to do that (use directly reg_Read, rmw or reg_Write) -in IPV_DMA: access only on 32 bits and use rmw or access only on 16 bits and use the doubled ralative macros for little and big endian Update exclusive areas if needed. To update the IPW access for APIs: Mcl_DmaTcdSetIterCount Mcl_DmaTcdSetLinkAndIterCount Mcl_DmaTcdGetIterCount Mcl_DmaUpdateIterCount Mcl_DmaGetCrtIterCount Mcl_DmaGetStartIterCount","'Files Modified: - Mcl_IPW.c - Mcl_Dma.c
ENGR00386442	NewWork	[MCL] Improve the DEM reporting, avoid code duplication 'NewWork Description: Improve the DEM reporting, avoid code duplication. Expected behavior: N/A Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): improve branch condition and reporting of dem report','Improve branch condition and reporting of dem report
ENGR00380918	NewWork	[MCL] Improve the TCD macro access 'NewWork Description: For accesing the TCD words on 32 bytes, MCL uses 2 sets of offset define which have the same value and the same meaning (word offset). Please replace the logical offsets with the word offests and delete the logical offsets: Example: DMA_TCD_TA_SOFF_OFFSET_U32 -

Table continues on the next page...

ID	Subtype	Headline and Description
		>DMA_TCD_2ND_WORD_OFFSET_U32. DMA_TCD_DLAST_SGA_OFFSET_U32 - >DMA_TCD_7TH_WORD_OFFSET_U32 DMA_TCD_DADDR_OFFSET_U32 ->DMA_TCD_5TH_WORD_OFFSET_U32 DMA_TCD_DOFF_OFFSET_U32-> DMA_TCD_6TH_WORD_OFFSET_U32 DMA_TCD_CITER_WORD_OFFSET_U32-> DMA_TCD_6TH_WORD_OFFSET_U32 DMA_TCD_TA_SOFF_OFFSET_U32-> DMA_TCD_2ND_WORD_OFFSET_U32 DMA_TCD_MLNO_OFFSET_U32-> DMA_TCD_3RD_WORD_OFFSET_U32 DMA_TCD_SLAST_OFFSET_U32 -> DMA_TCD_4TH_WORD_OFFSET_U32 Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please replace the logical offsets with the word offests and delete the logical offsets.","'Files Modified : - Reg_eSys_Dma.h - Mcl_Dma.c
ENGR00384818	NewWork	[MCL] Internal rework for the DMAMUX generated sources 'NewWork Description: Internal rework: move the generation part from Mcl_DmaMux.h to CDD_Mcl_Cfg.h. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Internal rework: move the generation part from Mcl_DMaMux.h to CDD_Mcl_Cfg.h","'Move the generated code for DMAMUX sources from Mcl_DmaMux.h to CDD_Mcl_Cfg.h Changed files: Mcl_Dmamux.h CDD_Mcl_Cfg.h Mcl.mak CDD_Mcl_PluginMacro.m delete symlink to Mcl_DmaMux.h(CC) or generated file Mcl_Dmamux.h (GIT)",ENGR00386424
ENGR00386902	Defect	[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)

Table continues on the next page...

ID	Subtype	Headline and Description
		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",'Files modified :,ENGR00387939
ENGR00384932	NewWork	 [MCL] READONLY attribute evaluated by XPATH should be replaced by EDITABLE attribute for AXBS parameters 'Problem detailed description (how to reproduce it): The MCL configuration data could not generated by command line if "Mcl Crossbar Switch Supported " is enabled. Tresos could not access resource file so it thrown error messges for XPATH in READONLY attribute of MclCrossbarPrioMasterX (X=0..7). e.g: Invalid XPath-expression for Attribute "READONLY" of node "/AUTOSAR/TOP-LEVEL-PACKAGES/Mcl/ELEMENTS/Mcl/MclConfigSet/MclConfigSet _0/MclCrossbarInstance/MclCrossbarInstance_0/MclCrossbarHwSlavePort/MclCrossbarH wSlavePort_0/MclCrossbarPrioMaster0": (35016) Cannot get a resource file for tar get PA and derivate MPC5777C. Maybe this is due to module restrictions. This is a Tresos issue. EB recommend to use EDITABLE attribute instead of READONLY in this case. There is similar issue in Fls driver tracked by ticket ENGR00383840 Preconditions: "Mcl Crossbar Switch Supported" and Tresos is generated by command line. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Generation error occured. Expected behavior: Configuration code is generated without error. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Implement workaround for Tresos issue: READONLY attribute is evaluated by XPATH accessing resource file should be replaced by EDITABLE attribute",'READONLY attribute is evaluated by XPATH

Table continues on the next page...

ID	Subtype	Headline and Description
		accessing resource file should be replaced by EDITABLE attribute.
ENGR00379347	NewWork	[MCL] Update de-intialize the master registers 'NewWork Description: Add in DeInit the part for to deinitialize the master registers used for Halo o #ifdef MCL_AXBS_ENABLE_INIT_MGPCR o #if (STD_ON == MCL_AXBS_ENABLE_INIT_MGPCR) Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","'File modified : - Mcl_Axbs.c
ENGR00384410	Defect	[MCL] Wrong range of DMAMUX channels in user manual 'Problem detailed description (how to reproduce it): User manual specifies: Configuration for DMA source slot in DmaMux0 (Physical DMA channels from 0 to 7) Configuration for DMA source slot in DmaMux1 (Physical DMA channels from 8 to 15) This is incorrect becasue DmaMux0 has assigned channels 0 to 15 and DmaMux1 has assigned channels from 16 to 31. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: User manual specifies: Configuration for DMA source slot in DmaMux0 (Physical DMA channels from 0 to 7) Configuration for DMA source slot in DmaMux1 (Physical DMA channels from 8 to 15) Expected behavior: User manual should specify: Configuration for DMA source slot in DmaMux0 (Physical DMA channels from 0 to 15) Configuration for DMA source slot in DmaMux1 (Physical DMA channels from 16 to 31) Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): it should be: Configuration for DMA source slot in DmaMux0 (Physical DMA channels from 0 to 15) Configuration for DMA source slot in DmaMux1 (Physical DMA channels from 16 to 31)",'Update UM,ENGR00384411

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00386022	NewWork	[MCU] Fix misra error 'NewWork Description: Fix misra error on RaceRunnerUltra Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fix misra error","'Mcu_AFE.c Mcu_CMU.c Mcu_Cfg.h Mcu_IPW.c Mcu_RegOperations.m
ENGR00385095	NewWork	[MCU] Implement CPR-MCAL-798.mcu 'NewWork Description: The MCU driver shall provide an optional MCU global parameter (e.g. McuPrepareMemoryConfig) to allow the user specifying the name of callout functions which has to be called before initializing the FLASH memory or before changing the FLASH and RAM memories parameters (e.g. wait states). The callout function shall be called by the Mcu driver with a parameter to specify if it is the entry point of the memory configuration or if is the exit point of the memory configuration, in order to stop or start the other cores than the one executing the MCU driver (e.g. HSM). Note: Rational: the other cores (e.g. HSM) which are not under AUTOSAR control need to be stopped for a short amount of time. The callout functions have to call just before and immediately after the functions changing the FLASH/RAM configuration (e.g. MCU_FLASH_Init() and MCU_Flash_SetWS() and Mcu_PRAM_SetRamWS()) Requirement source: CPRT requirement CPR-MCAL-798.mcu (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): A plugin parameter will be added to specify the name of the callout function and the FLASH and RAM memory initializations will be guarded by the call of that function specifying using a parameter the begging or ending of the initialization.","'Add McuPrepareMemoryConfig in Mcu.xdm Create a callout to be called in Mcu_IPW.c before and after configuring the flash and RAM controllers. Files updated: Mcu.xdm Mcu_IPW.c Mcu_Cfg.h design parameters.xml (IM) Mcu_MemoryConfigStageType.xml (UM) Mcu_GeneralConfiguration.xml (UM) AUTOSAR_MCAL_MCU_UM.ditamap
ENGR00387520	NewWork	[MCU] Review change according to new errata document for Calypso 3M/6M

Table continues on the next page...

ID	Subtype	Headline and Description
		RTM 1.0.3 ASR 4.0 'NewWork Description: - Review the list of erratum provided in "Log" sheet of each given reviewing template. - Fill the errata review template for your module with below rule: keeping only the erratum relating to your module and removing the others (not related to your module). Fill the information to column E, F, G, H, I in the template. Finally, attach the review result to this CR with this naming convention: SMCAL_Errata_review_<module>_C3M_cut2.0 or SMCAL_Errata_review_<module>_C3M_cut2.1 - The more errata information can be found in errata documents as mentioned in Requirement source. For Calypso 3M: target to support the latest cut 2.1 and 2.0 Some errata have been removed between the cuts and both cuts have had errata added. The added errata should be reviewed for affect in MCAL code. For Calypso 6M: target to support the latest cut 3.0 and 2.0 Calypso 6M - There are no new errata to be reviewed this time. All of them have been reviewed in the previous Calypso releases. There are only errata removed between the cuts at this time. Both 3M and 6M When moving from old cut to new cut, some of old erratum still exist in lastest cut (see in "Log" sheet of errata review template) but some others have been removed from latest cut (see in "Removed" sheet of errata review template). So, the modules which implemented workaround in code need to be reviewed again to see if we need to keep or remove the workaround. For Calypso 3M, the following modules need to be reviewed: PORT, DIO, MCL, MCU, SPI, GPT, FR, MCU, CAN and ADC. For Calypso 6M, the following modules need to be reviewed again: GPT, FR, MCU, CAN. Requirement source: Calypso 6M: Cut 2.0 MPC5748G_1N81M Mask Set Errata Rev .2, Jun2016 Calypso 3M cut 2.1: MPC5746C_Comparison_Summary_Ver0_0, 14-Sep-2016. Calypso 3M cut 1.1: MPC5746C_1N06M, Rev.4 July 2016 Calypso 6M cut 3.0: MPC5748G_1N81M_0N78S_Comparison_Summary_v2_0, 31-Oct-2016. Calypso 3M cut 2.1: C3M_cut2.1_new_errata_20170113 (internal document), 13-Jan-17 (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>_CxM_cutx.x. If any workaround is removed in the code this time, please take a note in "Remark" column to state that information. - 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)

Table continues on the next page...

ID	Subtype	Headline and Description
		+ Reporter (role): Developer/Designer + Link option: Add the New Feature ticket used for review with selection of "is a dependency for".','
ENGR00381759	NewWork	[MCU] Update code review against checklist 'NewWork Description: Update code according to finding in ENGR00379723 Requirement source: sMCAL Release criteria document: http://compass.freescale.net/go/228798570 Coding guideline: http://compass.freescale.net/go/227962899 Code review checklist: http://compass.freescale.net/go/224108405 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update code according to finding in ENGR00379723","'Mcu_EnvCfg.h Mcu_IPW_Irq.h Mcu_Cfg.h Mcu_IPW.c
ENGR00383261	NewWork	[MCU] Update code review against checklist for S32K14X RTM 1.0.0 'NewWork Description: Update code according to ENGR00382153 Requirement source: sMCAL Release criteria document version 25: http://compass.freescale.net/go/228798570 Coding guideline verion 5.0, July 2016: http://compass.freescale.net/go/227962899 Code review checklist verion 3.0, July 2016: http://compass.freescale.net/go/224108405 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update code according to ENGR00382153","'Mcu_IPW_Types.h Mcu.c Mcu_IPW.h Mcu_IPW.c
ENGR00385120	NewWork	[MCU] Update driver according to ASR4.2.2 specific on RR Ultra RTM release 'NewWork Description: Update the driver according to the requirements on ASR 4.2.2. The updated requirements have been identified using the ENGR00384031. 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:

Table continues on the next page...

ID	Subtype	Headline and Description
		gMRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Switch everywhere in the code the ASR version check. Update in Mcu.xdm McuClockSettingId, McuMode and the description of McuDevErrorDetect according to ASR 4.2.2","'Update the driver to support ASR 4.2.2 Files updated: Mcu.c Mcu.h Mcu.xdm Mcu_Cfg.h Mcu_Cfg.c Mcu_PBCfg.c Mcu_IPW.h Mcu_IPW_Irq.h Mcu_IPW.c All the files in all IPs",ENGR00385750 ENGR00385758 ENGR00385760 ENGR00385762 ENGR00385765 ENGR00385767 ENGR00385769 ENGR00385771 ENGR00386074 ENGR00386076 ENGR00386077 ENGR00386078 ENGR00386079 ENGR00386080 ENGR00386081
ENGR00381446	NewWork	[MCU] Verify all Misra errors and comments (part 2) 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fix the misra rule that can be fixed (see excel file in the CR ENGR00378353)",'N/A
ENGR00387807	Defect	[PORT] Correct the configuration of PCR32, PCR121 'Problem detailed description (how to reproduce it): The customer configured the PORT driver for Calypso 3M, the configuration of the PCR32, PCR121 for DCI signals are not correct. This causes an unexpected error that makes that uC enters in RAM corruption state. 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: the configuration for DCI signals are not generated correctly. e.g DCI_TDI with SMC checked. {(Port_InternalPinIdType)32, (uint32)0x00080000, (uint8)0, (uint8)0, (boolean)FALSE, (boolean)FALSE, (boolean)TRUE}, Expected behavior: the configuration for DCI signals are generated correctly. e.g DCI_TDI with SMC checked. {(Port_InternalPinIdType)32, (uint32)0x00880001, (uint8)0, (uint8)0, (boolean)FALSE, (boolean)FALSE, (boolean)TRUE}, 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 configuration of DCI_TDI_PORT32/DCI_TDI_PORT121 in Calypso_Resource_3M.m and generation of SMC bit to generate the expected data.",'
ENGR00382614	Defect	[PORT] Remove external declaration for variables in Port_Config structure from configuration header file and fix Memory Allocation Sections 'Problem detailed description (how to reproduce it): There are a number of variables in the source configuration file (ex: Port_au16NoUnUsedPadsArrayDefault, Port_UnUsedPin, Port_aPinConfigDefault, etc) which are part of the port configuration structure and do not need external access, as they are accessed by the driver through the pointer to the configuration structure. This ticket requires the following actions: 1. assure that all variables that are part of the configuration structure are made static inside the configuration source files and do not appear as external in header configuration file 2. assure that all variables defined and declared in the source and header configuration files are placed into the right MemMap section 3. assure that the variables that are not part of the configuration but are accessed by the driver are declared as external in configuration header file and are placed in the right MemMap section Preconditions: None Test Case ID (internal TC that caught the defect) - optional Dio_TS_013 Trigger: None Observed behavior: Linaro compiler raises error because a variable declared as static in source configuration file is also declared as external in the header configuration file. Expected behavior: No compiler errors generated. All variables in configuration files properly declared and defined and placed into the right 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): Create an excel file with the storage class of the variables in the configuration files and with their MemMap section. Highlight the errors. Use this ticket to implement the fixes. Please see attached file: Config_variables_<Platform>.xlsx",'Remove external declaration for variables in Port_Config structure from configuration header file

Table continues on the next page...

ID	Subtype	Headline and Description
		and correct the MemMap sections for the variables in configuration files
ENGR00383864	NewWork	[PORT] Update resource csv file to synchronize with reference manual on Calypso 3M 'NewWork Description: In the current, DCI_TDI pin and DCI_TCK pins on Calypso 3M can work normally with only setting IBE = ON, without setting SSS = 1 although in Calypso 3M documentation, those pins need to set both IBE to ON and SSS field to 1. However, Resource csv file should be updated to set SSS to 1 for those pins to avoid unexpected issues which may be appear. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update resource csv file to change the value of DCI_TDI and DCI_TCK from 9 to 1.",'Update resource csv file to change the value of DCI_TDI and DCI_TCK from 9 to 1.
ENGR00384566	NewWork	[PWM] Add support wireless charging lib and timeline 'NewWork Description: Add support Pwm_SyncUpdate Pwm_SetPeriodAndDuty_NoUpdate Pwm_SetDutyCycle_NoUpdate Create: Pwm_SetPhaseShift Pwm_SetPhaseShift_NoUpdate Pwm_EnableTrigger Pwm_DisableTrigger Pwm_MaskOutputs Pwm_UnMaskOutputs Pwm_ResetCounter Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA","'Pwm.xdm Pwm_Cfg.h Pwm.c Pwm.h Pwm_Ipw.c Pwm_Ipw.h
ENGR00378080	Defect	[PWM] Hardware specific defines should not be available in Pwm.h 'Problem detailed description (how to reproduce it): Hardware defines are available through: Pwm.h <- Pwm_Ipw_Types.h <- Pwm_Ftm_Types.h <- Reg_eSys_Ftm.h &

Table continues on the next page...

ID	Subtype	Headline and Description
		Ftm_Common_Types.h Users should not have acces to low level IP header files. Preconditions: Pwm.h is included in the User source or header files. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Users should not have acces to low level IP header files. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Do not include in Pwm_Ftm_Types.h headers with IP specific definitions.','Hardware specific defines are no longer available through Pwm.h
ENGR00382361	Defect	[PWM] The output signal is incorrectly generated for OPWMT with duty 50% 'Problem detailed description (how to reproduce it): We find an issue with some pwm output channel. I almost test all the pwm channel and find for some of them if we set the duty circle to 50%, it output 100% duty from the pin, but if we set as 49% or 51% or any other percent, it all work fine. Here list five output which have this issue. HW Pin Result PwmChannelConfigSet_FULL1_turn J3-72 nok PwmConf_PwmChannelConfigSet_FULL1_H1 J3-12 nok PwmConf_PwmChannelConfigSet_FULL1_G4 J3-5 nok PwmConf_PwmChannelConfigSet_FULL1_G1 J4-40 nok PwmConf_PwmChannelConfigSet_FULL1_F12 J2-12 nok Preconditions: The PWM channels are configured in OPWMT mode. Please see the configuration in the attachment. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The polarity is HIGH but the output signal for 50% are always LOW. Expected behavior: the output signal for 50% is generated correctly. 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: In the function Pwm_eMios_SetRegs_OPWMT_mode the following code is not neccesary for OPWMT mode: /* if (nTempDuty + Pwm_Offset == nPeriod) then u32TempDutyOffset will be 0 if the reference * channel is MCB/OPWFMB then we have [0x1, nPeriod] if we program regB with 0 a * compare event will never be generated. In this case we program the nPeriod value instead * and accept an error equal with 1 tick. */ if (0x0U == u32TempDutyOffset)

Table continues on the next page...

ID	Subtype	Headline and Description
		{ u32TempDutyOffset = nPeriod; } This piece of code shall be removed.",'Pwm_eMios.c
ENGR00387427	NewWork	[SPI] Create quality package for Calypso 3M/6M RTM 1.0.3 ASR 4.0 'NewWork Description: Provide the following documents: Traceability: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix.xls Traceability warning report: AUTOSAR_MCAL_<MODULE>_TraceabilityMatrix_Warnings.xls VSMD report: AUTOSAR_MCAL_<MODULE>_VSMDReport.html Integration manual: AUTOSAR_MCAL_<MODULE>_IM.pdf User manual: AUTOSAR_MCAL_<MODULE>_UM.pdf DOORs DXL check result: AUTOSAR_MCAL_<MODULE>_CheckDriverReq.txt UML design: AUTOSAR_MCAL_<MODULE>_SDD.eap Exclusive area report: AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls CheckDefReport: AUTOSAR_MCAL_<MODULE>_CheckDefsReport.xlsx. 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",'
ENGR00384757	NewWork	[SPI] Improve the configuration files 'NewWork Description: Improve the configuration files Spi_Cfg.h, Spi_Cfg.c, Spi_PBcfg.c Requirement source: Design/Dev (SW) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Create Spi_RegOperations.m file, File used by the C generation templates. This file contains macros used by the C generation templates. And in Spi_Cfg.h, Spi_Cfg.c, Spi_PBcfg.c will call those macros.","'Files affected: Spi_Cfg.h, Spi_PBcfg.c, Spi.mak New file: Spi_RegOperations.m
ENGR00384879	Defect	[SPI] Incorrect description of parameters SpiTimeClk2Cs and SpiTimeCs2Clk 'Problem detailed description (how to reproduce it): The description of some SPI parameters in the Tresos configuration schema is not accurate with the real functionality of HW/SW: We see the following problems: - the max. value of parameter SpiTimeClk2Cs (0.0001) does not match the description (Timing between clock and chip select - This parameter allows to use a range of values from 0 up to 0.00001 Sec. ...) - the description of parameter SpiTimeCs2Clk (Timing between chip select and clock) does not reflect the real behavior as mentioned in the picture attached

Table continues on the next page...

ID	Subtype	Headline and Description
		(Figure 28-39). The second occurrence of timing (tCSC) is not relevant for Chip Select - the description of parameter SpiTimeClk2Cs (Timing between clock and chip select) does not reflect the real behavior as mentioned in the picture attached (Figure 28-39). The timing (tASC) is not relevant for Chip Select Preconditions: NA Observed behavior: NA When can it be observed? (at configuration time, at runtime, at compile time?) NA Expected behavior: NA Proposed solution (Optional): Update the description of the parameters SpiTimeClk2Cs and SpiTimeCs2Clk in the Tresos plugin.","'Files Modified: - Spi.xdm - SpiExternalDevice_42.xml
ENGR00381509	NewWork	[SPI] Investigate Misra rule 11.4 appearances in the code 'NewWork Description: This CR raised for investigate Misra rule 11.4 appearances in the code, and used to fix misra 11.4. Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA","'Files affected: - Spi_DSPI.h - Spi_DSPI.c
ENGR00386278	Defect	[SPI] Job status after Spi_AsyncTransmit call 'Problem detailed description (how to reproduce it): When calling Spi_AsyncTransmit for Sequence with two jobs, only job status of first job in sequence changes to SPI_JOB_QUEUED/SPI_JOB_PENDING. Status for second job remains SPI_JOB_OK instead of changing to SPI_JOB_QUEUED. According to [SPI194] and [SPI038] state that the status of job accepted by Spi_AsyncTransmit should change to SPI_JOB_QUEUED. Preconditions: SPI driver is configured in Asynchronous mode. Several jobs in a sequence.

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: When calling Spi_AsyncTransmit for Sequence with two jobs, only job status of first job in sequence changes to SPI_JOB_QUEUED/SPI_JOB_PENDING When can it be observed? (at configuration time, at runtime, at compile time?) Runtime Expected behavior: The accepted jobs shall be SPI_JOB_QUEUED Proposed solution (Optional): NA","'Files affected: -Spi.c
ENGR00382627	Defect	[SPI] The harware module name is not correct for SPI0-SPI5 unit in the range of SpiPhyUnitMapping 'Problem detailed description (how to reproduce it): For spi unit we can select as SPIphyUnit parameter DSPI_0...DSPI_9 However in IO signal description in RM we have DSPI_0..DSPI_3 and SPI_0..SPI_5 Tresos SPIphyUNit IO Table DSPI_0 DSPI_0 DSPI_1 DSPI_1 DSPI_2 DSPI_2 DSPI_3 DSPI_3 DSPI_4 SPI_0 DSPI_5 SPI_1 DSPI_6 SPI_2 DSPI_7 SPI_3 DSPI_8 SPI_4 DSPI_9 SPI_5 Also in Port driver the signal names are DSPI_0..DSPI_3 and SPI_0..SPI_5. I think we should change the SPIphy parameter to be DSPI_0..DSPI_3 and SPI_0..SPI_5. (not DSPI_0..DSPI_9) Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The DSPI_4-DSPI_9 are seletable for SpiPhyUnitMapping. It imples DSPI_4=SPI_0, DSPI_5=SPI_1, DSPI_6=SPI_2, DSPI_7=SPI_3, DSPI_8=SPI_4, DSPI_9=SPI_5. Expected behavior: The SPI_0-SPI_5 in the selection for SpiPhyUnitMapping. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA","'Files affected: - Spi.xdm - Spi_Cfg.h - Spi_Cfg.c - Spi_Lcfg.c - Spi_PBcfg.c - all files in "spi\specific\resource" folder - im_isr_to_configure.xml - AUTOSAR_MCAL_SPI_UM.xml

Table continues on the next page...

ID	Subtype	Headline and Description
		- AUTOSAR_MCAL_SPI_IM.xml - add new files: Spi_Dspi_IsrTCF_SPI_0.xml, ... ,Spi_Dspi_IsrTCF_SPI_5.xml & Spi_Dspi_IsrRxDma_SPI_0.xml, ... ,Spi_Dspi_IsrRxDma_SPI_5.xml",ENGR00385796
ENGR00382961	Defect	[WDGIF] Function prototype not aligned with AUTOSAR standard 'Problem detailed description (how to reproduce it): WdgIf uses APIs Wdg_SetMode and Wdg_SetTriggerCondition with a const modifier for parameters and this does not match with the AUTOSAR definition. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional Wdg_TC_0914.c Trigger: NA Observed behavior: During compile/link with diab the following error messages occur when integrating SMCAL WDG with customer WDGIF: initializer type `unsigned char(*)(int const)' incompatible with object type `unsigned char(* const)(int) Expected behavior: No any error messages occur during compile/link 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 const modifier for Wdg_SetMode and Wdg_SetTriggerCondition parameters.",'Change type of array function pointer to reproduce customer error
ENGR00387796	Defect	[WDGIF] Incorrect file include structure of Watchdog Interface '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: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA 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.) 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

Table continues on the next page...

ID	Subtype	Headline and Description
		#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" #endif #endif #ifdef WDG_INSTANCE4 #if (WDG_INSTANCE4 == STD_ON) #include "Wdg_43_Instance4.h" #endif #endif #endif",'fix include file structure of WdgIf
ENGR00382960	Defect	[WDG] Function prototype not aligned with AUTOSAR standard 'Problem detailed description (how to reproduce it): APIs Wdg_SetMode and Wdg_SetTriggerCondition use a const modifier for parameters and this does not match with the AUTOSAR definition. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional Wdg_TC_0914.c Trigger: NA Observed behavior: During compile/link with diab the following error messages occur:During compile/link with diab the following error messages occur when integrating SMCAL WDG with customer WDGIF: initializer type `unsigned char(*)(int const)' incompatible with object type `unsigned char(* const)(int) Expected behavior: No any error messages occur during compile/link Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):

4.4 RTM 1.0.2

ID	Subtype	Headline and Description
ENGR00362211	Defect	[ADC] ADCLKSEL bit configuration is not restored to user settings after the calibration service is finished 'Problem detailed description (how to reproduce it): 1. The user configuration for ADCLKSEL bit is set in ADC registers in Adc_Init API. However, there is a specific configuration for that bit that is needed when executing the calibration API (the bit is set). The configuration of ADCLKSEL is should be restored to the actual value after Calibration is done, but this is not done, the bit remains set regradless of the user configuration. Workaround: If the user indends to use the Calibration API, they should take into account this limitation of the driver. If the user clock configuration for ADC has the ADCLKSET bit set, the clock confiugration will not be altered by the Calibration API. The user shoudl try to use this configuration for ADC clock settings. 2. Before calibration starts, ADCLKSEL should be configured as to reduce the frequency of ADC HW unit. It is recommended to have Calibration executed at max 40MHz, so clock settings internal to the ADC HW unit should be configured to produce the slowest clock.",'Corrected Calibration Api to restore user confiugration of ADCLKSEL bit
ENGR00369982	Defect	[ADC] ADC_EXCLUSIVE_AREA_07 exits twice in case timeout 'Problem detailed description (how to reproduce it): The customer reported following issue in MPC577XM MCAL4.0 FBR1.1.0: SchM_Exit_Adc_ADC_EXCLUSIVE_AREA_07 may be called twice. Adc_UpdateStatusStartConversion(L2152 Enter) -> Adc_Ipw_StopCurrentConversion -> Adc_Adcdigv2_StopCurrentConversion -> Adc_Adcdigv2_StopConversionCheckTimeout(L1699 Exit) (In case of timeout) -> Adc_UpdateStatusStartConversion(L2177 Exit)",'fix the defect that ADC_EXCLUSIVE_AREA_07 exits twice in case timeout
ENGR00378097	Defect	[ADC] ADC_TIMEOUT_COUNTER should be unsigned long 'Problem detailed description (how to reproduce it): ADC_TIMEOUT_COUNTER should be unsigned long because the node AdcTimeout is 32 bit integer. The code template below can raise the warning: "large integer implicitly truncated to unsigned type" #define ADC_TIMEOUT_COUNTER [!"num:inttohex(AdcGeneral/AdcTimeout, 8)"!]U It should be: #define ADC_TIMEOUT_COUNTER [!"num:inttohex(AdcGeneral/AdcTimeout, 8)"!]UL Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger:

Table continues on the next page...

ID	Subtype	Headline and Description
		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): #define ADC_TIMEOUT_COUNTER [!"num:inttohex(AdcGeneral/AdcTimeout, 8)"!]U It should be: #define ADC_TIMEOUT_COUNTER [!"num:inttohex(AdcGeneral/AdcTimeout, 8)"!]UL",'Update type of ADC_TIMEOUT_COUNTER to unsigned long
ENGR00363246	Defect	[ADC] AdcHwChannel: Range in epd-File not correct 'Problem detailed description (how to reproduce it): Found in version: MPC574XG_MCAL4_0_RTM_1_0_1 In Adc_mpc5746b_mapbga256.epd, the list of valid AdcHwChannel values is not the same like in PA_MPC574XG_mpc5746b_mapbga256.properties (e. g. AN_9). Please check all epd-Files Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: - Create a new list in the resource file that contain all channels of all hw unit like: Adc.AdcConfigSet.AdcHwUnit.AdcChannel.AdcHwChannel:AN_0,AN_1,AN_2,AN_3,AN_4,AN_5... And also implement this for other list - Modify Adc.xdm file to use above list for RANGE attribute and add a range check to ensure user select correct channel corresponding hw unit selected 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",'Correct list of valid AdcHwChannel in epd files
ENGR00377686	Defect	[ADC] AdcPriorityQueueMaxDepth should be precompile in all variants 'Problem detailed description (how to reproduce it): In Adc.xdm: AdcPriorityQueueMaxDepth should be precompile in all variants Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Expected behavior: AdcPriorityQueueMaxDepth should has precompile in all variant AdcPriorityQueueMaxDepth should be updated following code: <!-- AdcPriorityQueueMaxDepth --> <v:var name="AdcPriorityQueueMaxDepth" type="INTEGER"> <a:a name="DESC"> <a:v><![CDATA[EN:<html> Maximum depth of queue used for queuing of incoming conversion requests when hardware unit is busy. </html>]]></a:v> </a:a> <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PreCompile">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> <a:a name="LABEL" value="Adc Max Queue Depth"/> <a:a name="ORIGIN" value="M4_XDM_AR_MODULE_ORIGIN"/> <a:a name="SYMBOLICNAMEVALUE" value="false"/> <a:a name="UUID" value="ECUC:f80f20b5-00ab-43c2-9208-245774e7bc5f"/> <a:da name="DEFAULT" value="1"/> <a:da name="INVALID" type="Range"> <a:tst expr="<=65535"/> <a:tst expr=">=1"/> </a:da> <a:da name="RANGE" type="XPath"> <a:tst expr="../AdcPriorityImplementation = 'ADC_PRIORITY_NONE' and ../AdcEnableQueuing = 'false' and . !=1" true="The depth must be 1"/> </a:da> </v:var> 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",'AdcPriorityQueueMaxDepth has precompile in all variant
ENGR00380098	NewWork	[ADC] AdcPriorityQueueMaxDepth should be pre-compile 'NewWork Description: AdcPriorityQueueMaxDepth was moved into AdcGeneral, and the value is generated from AdcPriorityQueueMaxDepth used as pre-compile. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Adc_RegOperations.m: Remove the generate code for Adc_QueueMaxDepth element following the IPV. In IPV: The element Adc_QueueMaxDepth of struct: Adc_Adcdig_MultiConfigType, Adc_Eqadc_MultiConfigType, Adc_Adcdigv2_MultiConfigType should be removed. The reference to Adc_QueueMaxDepth: Adc_pCfgPtr->Misc.Adc_QueueMaxDepth will be replace by: ADC_QUEUE_MAX_DEPTH_MAX",'AdcPriorityQueueMaxDepth now is precompile,ENGR00380103

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00378093	NewWork	[ADC] Adc_SetMode and Adc_SetClockMode services should be independent 'NewWork Description: In Rayleigh and Kinetis, the function Adc_SetClockMode does not require to invoke SetMode function to POWER_DOWN_MODE. So two function is independence. The Invalid of node AdcEnableDualClockMode below should be remove from Adc.xdm: <a:da name="INVALID" type="XPath"> <a:tst expr="(node:fallback(.,'true') = 'true') and (node:fallback(../../AdcGeneral/AdcSetModeApi,'false') = 'false')" true="This paramter can be true only if AdcSetModeApi is enabled"/> </a:da> Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA",'Adc_SetMode and Adc_SetClockMode services should be independent,ENGR00381186
ENGR00379263	Defect	[ADC] Adc_ValidateNotBusyWithPrio function should be guarded under ADC_ENABLE_START_STOP_GROUP_API == STD_ON 'Problem detailed description (how to reproduce it): Adc_ValidateNotBusyWithPrio function is only called in function Adc_ValidateStateStartGroupConvNotBusy under the guard (ADC_ENABLE_START_STOP_GROUP_API == STD_ON). So the prototype of that function should be guarded under (ADC_ENABLE_START_STOP_GROUP_API == STD_ON) Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The prototype for should be guarded as below: #if ((ADC_PRIORITY_IMPLEMENTATION != ADC_PRIORITY_NONE) && (ADC_ENABLE_START_STOP_GROUP_API == STD_ON) || defined(__DOXYGEN__)) LOCAL_INLINE FUNC(Std_ReturnType, ADC_CODE) Adc_ValidateNotBusyWithPrio (VAR(Adc_GroupType, AUTOMATIC) Group, VAR(boolean, AUTOMATIC) bImplicitly); #endif /* ((ADC_PRIORITY_IMPLEMENTATION != ADC_PRIORITY_NONE) && (ADC_ENABLE_START_STOP_GROUP_API == STD_ON) || defined(__DOXYGEN__)) */ 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
		N/A",'Adc_ValidateNotBusyWithPrio function is guarded under ADC_ENABLE_START_STOP_GROUP_API
ENGR00372685	Defect	[ADC] Adc_VersionCheck_Src.m file in generate_PB folder is not signed 'Problem detailed description (how to reproduce it): In Adc_PBcfg.c, the following code section causes a problem: [!INCLUDE "Adc_VersionCheck_Src.m"!] Because Adc_VersionCheck_Src.m is not signed, Tresos reports an error during the code generation: "[.../Adc_TS_T2D32M20I0R0/generate_PB/include/Adc_PBcfg.c (signed)]: Failed to access include file "Adc_VersionCheck_Src.m" Preconditions: Usage of signed plugins and post build. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Generation error due to not signed file. Expected behavior: File to be signed in order to generate the code without error. 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",'Update Adc.mak
ENGR00378295	NewWork	[ADC] Add comment for compiler warning 'There are two compile warning raised by linaro toolchain: 2391 array subscript is above array bounds [-Warray-bounds]@47 Adc_UnitStatus[Unit].SwNormalQueue[QueueTemp] = (Adc_GroupType)Adc_UnitStatus[Unit].SwNormalQueue[QueueTemp- 1U]; 2475 array subscript is above array bounds [-Warray-bounds]@43 Adc_UnitStatus[Unit].SwNormalQueue[qPtr] = Group; They should be commented following the requirement: PR-MCAL-3205.adc: In case of a not avoidable compiler warning, this template for explanation shall be used for each occurrence inside the source code: /* Compiler_Warning: <reason to be provided> ... */",'Add comment for compiler warning on Linaro compiler
ENGR00381234	NewWork	[ADC] Add restrictions to the UM for the channel limit checking configuration for ASR 4.0.3 implementation 'NewWork Description: As Autosar specification for ADC, all limit checking parameters (AdcChannelLimitCheck, AdcChannelHighLimit, AdcChannelLowLimit, AdcChannelRangeSelect) are PreCompile in all variants but they are required to be included under AdcConfigSet/AdcHwUnit/AdcChannel so they can vary across configuration sets. Only the limit checking configuration from the first configuration will be used by Adc driver.

Table continues on the next page...

ID	Subtype	Headline and Description
		In order to maintain the consistency required by the PreCompile characteristic of these parameters, the user must follow a set of restrictions: - the same set of ADC units must be defined in all configurations, and ADC HW units must be mapped to the same ADC logical hardware unit ids - in all configurations, all Adc units should have the same number of channels configured - in all configurations, the limit checking parameters should be configured with the same values. Only the limit checking configuration from the first configuration will be used. These restrictions should be updated in ADC driver User Manual. Requirement source: ASR SWS (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update UM with restrictions for configuring limit checking parameters.",'Updated ADC UM with restrictions for configuring limit checking parameters.
ENGR00372816	NewWork	[ADC] Add support for ASR 4.2.1 'NewWork Description: Update configuration files according to ASR 4.2.1 standard, to work wih new EB Tresos version",'Add support for ASR 4.2.1 on Racerunner Ultra
ENGR00373935	Defect	[ADC] Add version check for Adc_Reg_eSys_Bctu.h in the Adc_Adcdig_Irq.c 'Problem detailed description (how to reproduce it): The code in Adc_Adcdig_Irq.c includes Adc_Reg_eSys_Bctu.h. But it does not check version mapping between Adc_Reg_eSys_Bctu.h and Adc_Adcdig_Irq.c Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The checking version should be added in Adc_Adcdig_Irq.c 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",'Add version check for Adc_Reg_eSys_Bctu.h in the Adc_Adcdig_Irq.c
ENGR00378780	Defect	[ADC] Avoid duplicated extern declaration of the notification prototypes 'Problem detailed description (how to reproduce it): In case of one notification is configured for different groups, that notification will have a duplicated prototype declaration in the Adc generated files. example: we configure <Notification> function for group0, group1, group2. ADC The prototype definition of Notification function in Adc_PBcfg.c will be generated as

Table continues on the next page...

ID	Subtype	Headline and Description
		below: extern void Notification(void); /**< @brief for Group - 0 */ extern void Notification(void); /**< @brief for Group - 1 */ extern void Notification(void); /**< @brief for Group - 2 */ Preconditions: the same notification function configured for more than one Adc group. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: the same notification function configured for more than one Adc group. Observed behavior: Extern prototype of function is declared more than once for the same function. Expected behavior: If the same notification is configured for difference group, it should have only one extern prototype declaration: extern void Notification(void); /**< @brief for Group - 0, 1, 2 */ 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",'Update to fix duplicate notification
ENGR00377097	NewWork	 [ADC] Calling to Mcl_DmaDisableNotification should be under MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON 'NewWork Description: By the current implementation of Mcl: #if (MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON) FUNC(void, MCL_CODE) Mcl_DmaEnableNotification (VAR(Mcl_ChannelType, AUTOMATIC) ChannelNumber, VAR(Mcl_DmaTransferNotifType, AUTOMATIC) Notification); FUNC(void, MCL_CODE) Mcl_DmaDisableNotification(VAR(Mcl_ChannelType, AUTOMATIC) ChannelNumber); FUNC(void, MCL_CODE) Mcl_DmaAcknowledgeInterrupt(VAR(Mcl_ChannelType,AUTOMATIC) ChannelNumber); #endif The calling to those function should be under #if (MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON)",'Calling to Mcl_DmaDisableNotification should be under MCL_DMA_NOTIFICATION_SUPPORTED == STD_ON
ENGR00381423	Defect	 [ADC] Configure ADCDIG_DMA_CLEAR_ON_READ in ADC_DMAE when starting an Adc group 'Problem detailed description (how to reproduce it): When Adc group is started and DMA transfer is used, ADCDIG_DMA_CLEAR_ON_READ should be set in ADC_DMAE register, otherwise there will be spurious triggers to DMA and incorrect data will be transferred. Preconditions: DMA transfer is configured Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Trigger: N/A Observed behavior: DMA transfers data before the conversions are complete Expected behavior: DMA transfers data before only after the conversions are complete Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Revert the setting for DMA clear on read",'Revert ADCDIG_DMA_CLEAR_ON_READ configuration bit.
ENGR00381123	NewWork	[ADC] Create Adc_Adcdig_Irq.h to contain function's prototype 'NewWork Description: Some function's prototype is put on source file. The function's prototype should be put in header file follow C standard Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Create Adc_Adcdig_Irq.h to include prototype for Adc_Adcdig_EndMultipleCtuConv and Adc_Adcdig_EndHardwareConv",'Create Adc_Adcdig_Irq.h to contain function's prototype,ENGR00381129
ENGR00371520	NewWork	[ADC] Ctu trigger mode should be guarded for platform using ctuv2 'Ctu trigger mode is not supported for RaceRunner IS. We should creat a precompile define: ADC_CTU_TRIGGER_MODE_SUPPORTED = OFF. Using this precompile to guard code in IPV.",'RaceRunner IS is not supported CTU trigger mode.,ENGR00371523
ENGR00363185	Defect	[ADC] CurrentChannel should be aligned with Adc_ChannelIndexType 'CurrentChannel element of Adc_GroupStatusType now is Adc_ChannelType (8 bit) incase of group is configured 255 channel, it can be overflow when CurrentChannel is updated. So CurrentChannel element of Adc_GroupStatusType should be aligned with Adc_ChannelIndexType (16 bit) Problem detailed description (how to reproduce it): N/A (description of the defect ? what is not correct, what errors/warnings are there, where are they, etc...) Preconditions: N/A (what is needed to trigger the bug) Expected behavior: N/A (how it should behave)",'Change the type of element CurrentChannel in Adc_GroupStatusType by Adc_ChannelIndexType

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00371799	Defect	[ADC] Don't need to configure Channel interrupt mask register for Adc conversion 'Problem detailed description (how to reproduce it): The End of Conversion interrupt is not used by current implementation. So the driver does not needs to configure Channel interrupt mask register for Adc conversion. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Remove in resource Channels interrupt mask register, remove also in the template file. Remove au8Adc_CimrPresent[ADC_MAX_HW_UNITS][ADC_CIMR_REGS]; /**< @brief CIMR registers working for each unit */ in the Adc_Adcdig_CfgEx.h. Remove function Adc_Adcdig_SetCimrRegisters in Adc_Adcdig.c Remove define for Channel Interrupt Mask Register (CIMR0-2) in Adc_Reg_eSys_Adcdig.h 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",'The End of Conversion interrupt is not used by current implementation. So the driver don't needs to configure Channel interrupt mask register for Adc conversion.,ENGR00371838
ENGR00378230	Defect	[ADC] Duplicate prototype for Adc_ValidateExtraParams function (redundant code) 'Problem detailed description (how to reproduce it): In Adc.c file there are two prototypes for Adc_ValidateExtraParams function and the function prototype and function definition are not synchronized. See below for more details: The first prototype: #if ((ADC_VALIDATE_PARAMS == STD_ON) && \ ((ADC_ENABLE_START_STOP_GROUP_API == STD_ON) || \ (ADC_HW_TRIGGER_API == STD_ON) || \ (ADC_ENABLE_CTUTRIG_NONAUTO_API ==STD_ON) || \ (ADC_ENABLE_CH_DISABLE_CH_NONAUTO_API == STD_ON) \ ) \ ) LOCAL_INLINE FUNC(Std_ReturnType, ADC_CODE) Adc_ValidateExtraParams (VAR(uint8, AUTOMATIC) u8ServiceId, VAR(uint32, AUTOMATIC) u32ErrorIdList, VAR(Adc_GroupType, AUTOMATIC) Group); #endif The second prototype: #if ((ADC_ENABLE_START_STOP_GROUP_API == STD_ON) || \

Table continues on the next page...

ID	Subtype	Headline and Description
		(ADC_HW_TRIGGER_API == STD_ON) || \ (ADC_ENABLE_CTUTRIG_NONAUTO_API ==STD_ON) || \ (ADC_ENABLE_CH_DISABLE_CH_NONAUTO_API == STD_ON) \ ) LOCAL_INLINE FUNC(Std_ReturnType, ADC_CODE) Adc_ValidateExtraParams (VAR(uint8, AUTOMATIC) u8ServiceId, VAR(uint32, AUTOMATIC) u32ErrorIdList, VAR(Adc_GroupType, AUTOMATIC) Group); #endif The function definition: static FUNC(Std_ReturnType, ADC_CODE) Adc_ValidateExtraParams (VAR(uint8, AUTOMATIC) u8ServiceId, VAR(uint32, AUTOMATIC) u32ErrorIdList, VAR(Adc_GroupType, AUTOMATIC) Group) { ... } Preconditions: n/a Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Adc_ValidateExtraParams will have only one, correct prototype. 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",'Correct the function definition of Adc_ValidateExtraParams
ENGR00372309	Defect	[ADC] Duplicate the define ADC_UNIT_x_ISR_USED 'Problem detailed description (how to reproduce it): The define for ADC_UNIT_X_ISR_USED is generated duplicate in Adc_CfgDefines.h when it has more than one configuration for one ADC hardware. See the generate code from Adc_CfgDefines.h below for more detail: /** * @brief IRQ definition, the implementation for PR-MCAL-3134 */ [!LOOP "AdcConfigSet/*/AdcHwUnit/*"![!// [!NOCODE![!// [!VAR "AllGroupWithoutInterrupt" = "1"!] [!LOOP "./AdcGroup/*"!] [!IF "AdcWithoutInterrupts = 'false'"!] [!VAR "AllGroupWithoutInterrupt" = "0"!] [!ENDIF!] [!ENDLOOP!] [!ENDNOCODE![!/

Table continues on the next page...

ID	Subtype	Headline and Description
		[!IF "$AllGroupWithoutInterrupt = 0"!][!// #define ADC_UNIT_[!"substring-after(AdcHwUnitId, 'ADC')"!]_ISR_USED [!ENDIF!][!// [!NOCODE!][!// [!VAR "AdcWdgFeatureUsed" = "0"!] [!LOOP "./AdcChannel/*"!] [!IF "AdcEnableThresholds = 'true'"!] [!VAR "AdcWdgFeatureUsed" = "1"!] [!ENDIF!] [!ENDLOOP!] [!ENDNOCODE!][!// [!IF "$AdcWdgFeatureUsed = 1"!][!// #define ADC_UNIT_[!"substring-after(AdcHwUnitId, 'ADC')"!]_WDG_ISR_USED [!ENDIF!][!// [!ENDLOOP!][!// When one hardware trigger is configured for more than one configset. Example ADC0 is configured in Configset0 and Configset1. The define ADC_UNIT_0_ISR_USED will be generated two time. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The generate code should be fixed into: /** * @brief IRQ definition, the implementation for PR-MCAL-3134 */ [!VAR "maxUnit"="num:i(ecu:get('Adc.AdcConfigSet.AdcHwUnit'))-1"!][!// [!FOR "Unit" = "0" TO "num:i($maxUnit)"!][!// [!VAR "AllGroupWithoutInterrupt" = "1"!] [!VAR "AdcWdgFeatureUsed" = "0"!] [!LOOP "AdcConfigSet/*/AdcHwUnit/*[AdcHwUnitId = concat('ADC', $Unit)]"!][!// [!NOCODE!][!// [!LOOP "./AdcGroup/*"!] [!IF "AdcWithoutInterrupts = 'false'"!] [!VAR "AllGroupWithoutInterrupt" = "0"!] [!ENDIF!] [!ENDLOOP!] [!LOOP "./AdcChannel/*"!] [!IF "AdcEnableThresholds = 'true'"!] [!VAR "AdcWdgFeatureUsed" = "1"!] [!ENDIF!] [!ENDLOOP!] [!ENDNOCODE!][!// [!ENDLOOP!][!// [!IF "$AllGroupWithoutInterrupt = 0"!][!// #define ADC_UNIT_[!"$Unit"!]_ISR_USED [!ENDIF!][!// [!IF "$AdcWdgFeatureUsed = 1"!][!// #define ADC_UNIT_[!"$Unit"!]_WDG_ISR_USED [!ENDIF!][!// [!ENDFOR!] [...]

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): N/A",'Fix compiler warning from doplicate define for ADC_UNIT_X_ISR_USED
ENGR00379196	NewWork	[ADC] Fix code review findings 'NewWork Description: There are some findings after perform code review against checklist. The report is attached in the ENGR00377412. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please use the output of ENGR00377412 in attachment to update the code according the findings.",'Update after code checklist review,ENGR00379751 ENGR00379754
ENGR00371385	Defect	[ADC] Fix code review findings against checklist 'Problem detailed description (how to reproduce it): There are some findings after code review against checklist from CR:ENGR00370863 Fix that findings follow the attach file from CR: ENGR00370863 Preconditions: N/A 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",'Fix findings code review against checklist for RacerunnerIS RTM 2.0.0,ENGR00371395
ENGR00374024	Defect	[ADC] Fix compiler warning 'Problem detailed description (how to reproduce it): There are some warning in Adc_Adcdig.c in label: INT_IPV_ADCDIG_SMCAL_4.0_02.18.01_I03 1781 warning #826-D: parameter "Group" was never referenced@36 VAR(Adc_GroupType, AUTOMATIC) Group, 3057 warning #550-D: variable "TempCheck" was set but never used@37 VAR(Std_ReturnType, AUTOMATIC) TempCheck = (Std_ReturnType)E_OK; Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		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","'Guard the variable "TempCheck" in function Adc_Adcdig_Calibrate by M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001 Remove the input variable "Group" is not used in function Adc_Adcdig_PrepareGroupStart
ENGR00378337	NewWork	[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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		- 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",'Implement data consistency mechanism (exclusive areas)
ENGR00368160	NewWork	[ADC] 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.",'Update description for ADC_E_QUEUE and ADC_E_PARAM in User Manual
ENGR00365654	NewWork	[ADC] Improve generation code to get Dma channels 'NewWork Description: Currently, the generation code to get Dma channels for Dma is not generic, it's belong to platform. I see it can be improved by the way make it become generic. We can found the DmaSource from DmaHwChannel and take the right value for MclDMAChannelId. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): We can found the DmaSource from DmaHwChannel and take the right value for MclDMAChannelId. The code to generate MclDMAChannelId can be implemented follow: [!IF "AdcGeneric/AdcTransferType = 'ADC_DMA'"![!// [!NOCODE![!// [!FOR "Unit" = "0" TO "num:i($maxUnit)"![!// [!VAR "foundDma"="0"![!// [!VAR "mcl_Id"="0"![!// [!LOOP "AdcHwUnit/*"![!// [!VAR "temp" = "num:i(substring-after(string(AdcHwUnitId),'ADC'))"![!//

Table continues on the next page...

ID	Subtype	Headline and Description
		[!"$temp"!][!// [!LOOP "node:refs('ASPathDataOfSchema:/M4_XDM_AR_PKG_NAME/Mcl')/ MclConfigSet/*[1]/DMAChannel/*"!][!// [!IF "$Unit = $temp"!][!// [!VAR "temp2"="node:value(string(concat('DmaSource',string(num:i(num:i(substring- after(DmaHwChannel,'eDMA_')) div 16)))))"!][!// [!IF "$temp2 = concat('ADC',$Unit)"!][!// [!VAR "foundDma"="1"!][!// [!VAR "mcl_Id"="string(MclDMAChannelId)"!][!// [!BREAK!][!// [!ENDIF!][!// [!ENDIF!][!// [!ENDLOOP!][!// [!ENDLOOP!][!// [!CODE!][!// [!IF "$foundDma = 1"!][!// (uint8)[!"$mcl_Id"!]U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the DMA channel number for ADC [!"$Unit"!] */ [!ELSE!] (uint8)255U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the DMA channel number for ADC [!"$Unit"!] */[!// [!ENDIF!][!// [!ENDCODE!][!// [!ENDFOR!][!// [!ENDNOCODE!][!// [!ELSE!] [!FOR "Unit" = "0" TO "num:i($maxUnit)"!][!// (uint8)0U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] [!ENDFOR!][!// [!ENDIF!][!// },','Improve generation code to get Dma channels
ENGR00372658	NewWork	[ADC] Improve performance of DMA tranfer by using scatter-gather feature of DMA hardware 'NewWork Description: Improve performance of DMA transfer with scatter-gather configuration for groups with gaps: the current implementation for groups with gaps transfers data for all channels from the first to last into an intermediary buffer, then this data is re-arranged in the user defined buffer in the Dma notification. this is not efficient and can also cause hardware errors when accessing the cahnnels in the gaps, if the channels are not valid (the associated data registers are not defined). This can be improved by using scatter-gather feature of DMA hardware. The group can be split into sub-segments of continuous channels, and for each segment a scatter-gather TCD configuration will be defined. the group conversion will consist of executing all the TCDs. this is done without software intervention, the DMA transfers the next TCD configurations automatically. Only one interrupt occurs for a group conversion, after the last TCD is finished. Expected behavior: N/A 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): N/A",'Improverment DMA transfer by using scatter-gather feature,ENGR00372947
ENGR00378743	NewWork	[ADC] Improve xdm file to more generic by using resource file 'Problem detailed description (how to reproduce it): In node: AdcLogicalUnitId <a:da name="INVALID" type="Range"> <a:tst expr="<=1"/> <a:tst expr=">=0"/> </a:da> Some platform has more than two hardware unit. The Range of INVALID attribute should be corrected Review again xdm file and check what can be improve to generic. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: example for Matterhorn, the number of hardware unit is 12 so AdcLogicalUnitId should be updated as below: <a:da name="INVALID" type="Range"> <a:tst expr="<ecu:get('Adc.AdcConfigSet.AdcHwUnit')"/> <a:tst expr=">=0"/> </a:da> 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",'Improve xdm file to more generic
ENGR00323645	NewWork	[ADC] Improvements for configuration files - create macros and macro files 'NewWork Description: A great part of the generated code in ADC module is platform independent. The maintainance and overall size of this code can be improved by moving the generic parts into a generic file containing macros. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The maintainance and overall size of this code can be improved by moving the generic parts into a generic file containing macros.",'Moving the generic parts of generate files into a generic file containing macros
ENGR00368572	Defect	[ADC] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections

Table continues on the next page...

ID	Subtype	Headline and Description
		'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];','Correct memory allocation,ENGR00368933 ENGR00368941
ENGR00373408	Defect	[ADC] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in our MCAL drivers. Variable declarations should have the definitions and all extern declarations

Table continues on the next page...

ID	Subtype	Headline and Description
		mapped to the same memmap sections. See attached file for all rules specifying the use of Memmap sections. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Global variables used by Mcal drivers are mapped to default .data sections. Expected behavior: All global variables used by Mcal drivers must be mapped to the special .mcal_data sections. 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",'[ADC] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2)
ENGR00365665	Defect	[ADC] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): 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 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.mPreconditions: Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: 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

Table continues on the next page...

ID	Subtype	Headline and Description
		- ArPatchVersion with ArReleaseRevisionVersion 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.mPreconditions: 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",'Correct version checking in configuration template files
ENGR00369089	NewWork	[ADC] Interrupt / Dma transfer should be configurable for each Adc unit 'NewWork Description: Interrupt / Dma transfer should be configurable for each Adc unit- currently this is configured globally. a new parameter for interrupt or dma transfer should be added in the Adc unit configuration and checked whenever a group is started or stopped, and also in the interrupt routines. Expected behavior: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA",'Interrupt / Dma transfer should be configurable for each Adc unit,ENGR00369967 ENGR00380963
ENGR00373898	NewWork	[ADC] Investigate and implement to reduce cyclomatic complexity and nesting level 'NewWork Description: 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. 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. 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. 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

Table continues on the next page...

ID	Subtype	Headline and Description
		4. The requirements can be marked as Fulfilled In in the module has the maximum nesting level lower than 4 ---------- Proposed solution (Optional): In AUTOSAR_MCAL_<MDL>_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments - for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth >4 the problems shall be resolved",'Seperate the functions Adc_UpdateStatusAfterReadGroup and Adc_ValidateStateNotBusy into many function to decrease the cyclomatic complexity.
ENGR00373936	NewWork	[ADC] Keep DMA request enable at the end of DMA transmission 'Initial Description: Currently, at the end of the major loop, the DMA request will be disabled due to DREQ bit of TCD_CSR register is set. The customer is requesting to configure DREQ=0 so that the DMA request will not be disabled. The background is that They do not need to use Adc group Status so the customer does not want the interrupt at the end of DMA transmission. CE's comment: 1) Add new parameter like Always enable DMA request. 3) if "Always enable DMA request" is checked, does not set DREQ bit of TCD_CSR in Adc_Adcdigv2_StartDmaOperation.","'Does not set DREQ bit of TCD_CSR in Adc_Adcdigv2_StartDmaOperation Remove DMA_TCD_DISABLE_REQ from Mcl_DmaTcdSetFlags calling. In the Adc_Adcdigv2_DmaEndNormalConv and Adc_Adcdigv2_DmaEndHardwareConv, remove Mcl_DmaEnableHwRequest calling
ENGR00369107	Defect	[ADC] Misra rule 10.3 violation 'Problem detailed description (how to reproduce it): In the Adc_ReadGroup function, a comparison of 2 parameters have different type is violated misra rule 10.3. They should be cast to the same type before comparison. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Cast the 2 parameters in below code statement to the same type: if((GroupRet == E_OK) || (bFlag == FALSE))

Table continues on the next page...

ID	Subtype	Headline and Description
		Fixed code: if((GroupRet == (Std_ReturnType)E_OK) || (bFlag == (boolean)FALSE)) Preconditions: 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",'Fix missra violation on Adc.c and Adc_PBcfg.c
ENGR00362689	Defect	[ADC] Non-AUTOSAR References are wrong 'Problem detailed description (how to reproduce it): Non-AUTOSAR references are wrong in Module Description. Reference: <a:da name="REF" value="ASPathDataOfSchema:/AUTOSAR/EcucDefs/Adc/AdcConfigSet/AdcHwUnit/AdcThresholdControl"/> AdcThresholdControl: that is hw feature, not autosar specific. For example: 'McuPerRunConfig' <DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/AUTOSAR/EcucDefs/Mcu/McuModuleConfiguration/McuRunConfig</DESTINATION-REF> must be <DESTINATION-REF DEST="ECUC-PARAM-CONF-CONTAINER-DEF">/TS_T2D17M9I0R0/Mcu/McuModuleConfiguration/McuRunConfig</DESTINATION-REF> This is also wrong for 'McuPerLowPwrConfig' Preconditions: NA Reported release baseline: BLN_SMCAL_4.0_PANTHER_RTM_1.0.0 Observed behavior: Generation error when customer's tool (other than EB Tresos) is used When can it be observed? (at configuration time, at runtime, at compile time?) at configuration time Expected behavior: NA Proposed solution (Optional): [...]",'Correct non-autosar reference parth
ENGR00370925	Defect	[ADC] Normal hardware trigger is not disabled in Adc_DeInit 'Problem detailed description (how to reproduce it): The customer reported following defect in MPC577XM MCAL4.0 FBR1.1.0 The Normal hardware trigger is not disabled in Adc_DeInit. The function Adc_Adcdig_DeInitHardware does not clear SIUL2 registers for Normal hardware trigger. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		function Adc_Adcdig_DeInitHardware should be update as below: after line of code: REG_WRITE32(ADCDIG_PDEDR_REG_ADDR32(Unit), ADCDIG_PDEDR_RESET_VALUE_U32); Disable siul2 for normal conversion #if (STD_ON == ADC_HW_TRIGGER_API) #if (ADC_HARDWARE_NORMAL_CONVERSIONS_USED == STD_ON) /* Disable Hardware Trigger Selection */ Adc_Siul2_DisableHwTrigger(Unit); #endif /* (ADC_HARDWARE_NORMAL_CONVERSIONS_USED == STD_ON) */ #endif /* (STD_ON == ADC_HW_TRIGGER_API) */ [...] 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",'Fix the defect that Normal hardware trigger is not disabled in Adc_DeInit
ENGR00375447	NewWork	[ADC] Optimize RAM consumption 'NewWork Description: Optimize RAM consumption - currently global variables are defined per unit, for all available units. Defining them just for the units used in the driver configuration would save some RAM memory. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. Move the static configuration for WDG channel description to configuration data (Adc_Adcdigv2_ChannelConfigurationType) 2. Create new macro that defines number of hardware units configured and max hw channel id configured. Declare the Dma internal buffer according to that defines.",'Optimize RAM consumption,ENGR00376223
ENGR00376031	NewWork	[ADC] Refactoring the code to use logical id and physical id consistently 'NewWork Description: The configuration data can contains an array to mapping between logical Id and physical Id to remove some for loop in the code for increasing performance Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): In the Adc_Adcdig_MultiConfigType structure definition, add a new Adc_HwUnitType array with maximum hardware unit element as below: CONST(Adc_HwUnitType, ADC_CONST) aHwLogicalId[ADC_MAX_HW_UNITS]; It's value will be generated in the Adc_Structure_Config macro. The value for units that are not configured will be invalid value (255) In the code, all for loop to get the index of hw unit configuration structure will be replaced by:

Table continues on the next page...

ID	Subtype	Headline and Description
		HwIndex = Adc_CfgPtr->Misc.aHwLogicalId[Unit]; Update UM to adapt to the change in the Adc_Adcdig_MultiConfigType.",'Refactoring the code to use logical id and physical id consistently,ENGR00379990
ENGR00372109	NewWork	[ADC] Remove eStatus element from Adc_UnitStatusType 'NewWork Description: The element eStatus in Adc_UnitStatusType only used one time in Adc.c. It can be removed and the checking eStatus can be changed by checking queue index. Expected behavior: N/A Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Remove eStatus element in Adc_UnitStatusType,ENGR00372110
ENGR00378768	Defect	[ADC] Remove the element GroupAction not be used 'Problem detailed description (how to reproduce it): The element GroupAction in Adc_GroupConfigurationType is not used in any where, so it should be removed Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The generate code in macro Adc_Group_Config in Adc_RegOperations.m should be updated: Remove below code: [!IF "AdcGroupAction = 'ADC_NORMAL_CONVERSION' "!][!// /**< @brief ADC_NORMAL_CONVERSION */ (uint8)0x0, [!ELSEIF "AdcGroupAction = 'ADC_SELF_TEST_OPEN_CIRCUIT' "!][!// /**< @brief ADC_SELF_TEST_OPEN_CIRCUIT */ (uint8)0x1, [!ELSE!][!// /**< @brief ADC_SELF_TEST_SHORT_CIRCUIT */ (uint8)0x2, [!ENDIF!][!// The type of Adc_GroupConfigurationType in Adc_Adcdigv2_CfgEx.h should be updated: Remove below element: CONST(AdcGroupActionType, ADC_CONST) GroupAction; /**< @brief Group's type of conversion: regular or self test */ 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",'Remove GroupAction element from Adc_GroupConfigurationType
ENGR00372700	Defect	 [ADC] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. In Adc.mak, there is hard-coded copyright: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Hard-coded copyright does not exixst in plugin files. 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","updated Adc.mak From: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics To: (c) Copyright M4_SRC_YEAR_ID M4_SRC_COPYRIGHTED_TO Files Modified: Adc.mak
ENGR00371996	Defect	 [ADC] Result of previous conversion round are overwritten and unit status is not correct 'Problem detailed description (how to reproduce it): -When a group is aborted by a higher priority group, data of previous conversion rounds are overwritten. -If call function Adc_StopGroupConversion after group's state is changed to STREAM_COMPLETED (Oneshot/ Linear Streaming), the state of unit is ADC_IDLE even there are some groups remain in queue Procedure: Add 3 groups: Group 9, Single Continuous, SW trigger Group 10, Cicurlar Streaming, SW trigger Group 11, Linear Streaming, SW trigger Steps: -Start Group 9, call start Group 10 when group 9 has 5 completed samples. -Call start group 11 when group 10 has 5 completed samples -Call Adc_GetGroupStatus until group 11 changes state to STREAM_COMPLETED -Disable interrupt -Stop Group 11 -Check state of unit still at BUSY

Table continues on the next page...

ID	Subtype	Headline and Description
		-Check state, variable notification of group 9, group 10 -Store result of group 10 in an array -Enable interrupt -Call Adc_GetGroupStatus until group 10 changes state to STREAM_COMPLETED -Disable interrupt -Stop Group 10 -Check state of unit still at BUSY -Check state, variable notification of group 9 -Check the 5 first samples of group 10 are not overwritten -Enable interrupt -Call Adc_GetGroupStatus until group 19 changes state to STREAM_COMPLETED -Disable interrupt -Stop Group 9 Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional Adc_TC_0036 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",'Fix the error result is overwritten when one group is aborted by other group with higher priority,ENGR00372026
ENGR00372147	Defect	[ADC] Software inject group is aborted when stop normal conversion group 'Problem detailed description (how to reproduce it): In case we have a software normal and software injected group work in parallel. If we call Adc_StopGroupConversion function for software normal group. Both software normal and software injected group will be stopped. RM said: When an ABORTCHAIN is requested while an injected chain is running over a suspended normal chain, both injected and normal chains are aborted; both MSR[NSTART] and MSR[JSTART] are reset. ABORT and ABORTCHAIN requests are ignored during CTU conversion. ABORT requests are also discarded during Self-Test conversion. Preconditions: N/A 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

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",'Raise error when Software inject group is aborted while stop normal conversion group
ENGR00373357	NewWork	 [ADC] Some Non-Autosar confiugration parameters should have OPTIONAL attribute 'Summary the discussion between TL and CE: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). NewWork Description: Implementation details The parameters that can be excluded from a specific configuration shall be marked as optional (<a:a name="OPTIONAL" value="true"/>, which results in lower multiplicity =0 and upper multiplicity 1). The AUTOSAR parameters have already defined the lower and the upper multiplicity. This change is needed only for the NON-Autosar parameters. The boolean Non-Autosar parameters that enable / disable Non-Autosar features must not be marked as OPTIONAL - only the additional configuration parameters that are needed only when Non-Autosar features are enabled. All parameters that are under this category shall be marked as OPTIONAL=true in the xdm. The check that the features were correctly configured shall be transferred to the boolean parameter that enables the feature. Example: Node1 - configuration node for Non-Autosar feature that selects between transfer mode interrupt or DMA should not be OPTIONAL Node2 - configuration node that selects Dma channel reference needed when DMA transfer is selected should be OPTIONAL when Node1 selects DMA, there should be added an invalid check for the existence of Node2 Reference point SpiPhyTxDmaChannel will be enabled when SpiPhyUnitAsyncMethod is set up to be DMA the changes: into <v:var name="SpiPhyUnitAsyncMethod" type="ENUMERATION"> [...to be added the check that everything is configured the INVALID tag] <a:da name="INVALID" type="XPath"> <a:tst expr="((node:value(.)='DMA') and (node:exists(../SpiPhyTxDmaChannel))) or not(node:value(.)='DMA')" false="Please enable SpiPhyTxDmaChannel"/> </a:da> <v:ref name="SpiPhyTxDmaChannel" type="CHOICE-REFERENCE"> [...] <a:a name="OPTIONAL" value="true"/> <a:da name="EDITABLE" type="XPath" expr="not(node:value(../SpiPhyUnitAsyncMethod) = 'DMA')"/> ---------- Original request from customer engineer---------- Parameters: - Non-autosar parameters. - Specific parameters under certain conditions.

Table continues on the next page...

ID	Subtype	Headline and Description
		This is my additional details: Customer don't use some reference nodes (DMA with Spi, Clock reference with GPT...). Epc file that exported from EB tresos with no reference value in DMA SPI case. For ex: <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyRxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannelAux</DEFINITION-REF> </ECUC-REFERENCE-VALUE> But in EPD file, these parameters are defined as mandatory with lower/upper multiplicity =1. So they should be have default value because a mandatory parameter cannot be empty. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The not used parameters _should_not_ be part of the Tresos Export. So please change the xdm-file in that way that the parameters will not be part of the export for the case that they are not used! 2. Lower mu",'Added OPTIONAL attributes in xdm configuration files for NonAutosar nodes that are not mandatory.
ENGR00380515	Defect	[ADC] The driver shall check group status for only the group in current configuration 'Problem detailed description (how to reproduce it): The function Adc_DeInit calls Adc_ValidateDeInitNotBusy function which checks group status of ADC_MAX_GROUPS. The customer use-case has two configuration set with different number of group in each configuration set. When they calls Adc_DeInit for the configuration with lower number of Adc group. The function returns DET error. Preconditions: There are two configuration sets with different number of group. Observed behavior: DET error When can it be observed? (at configuration time, at runtime, at compile time?) at runtime Expected behavior: No DET error Proposed solution (Optional): In Adc_ValidateDeInitNotBusy, replace: for (GroupId = 0U; GroupId < (Adc_GroupType)ADC_MAX_GROUPS; GroupId++)

Table continues on the next page...

ID	Subtype	Headline and Description
		by: for (GroupId = 0U; GroupId < (Adc_GroupType)Adc_CfgPtr->GroupCount; GroupId++)",'The driver shall check group status for only the group in current configuration
ENGR00371116	Defect	[ADC] The driver should refers to the DMA channels that are enabled from Mcl only 'Problem detailed description (how to reproduce it): The bellow code refers to all DMA channels that are configured even if they are not enabled and usable. [!LOOP "AdcHwUnit/*"!] [!VAR "temp" = "string(substring-after(string(AdcHwUnitId),'SARADC_UNIT_'))"!] [!LOOP "node:refs('ASPathDataOfSchema:/TS_T2D36M11I0R0/Mcl')/MclConfigSet/*[1]/DMAChannel/*"!] [!IF "$Unit = $temp"!] [!IF "num:i(substring-after(DmaHwChannel,'eDMA_')) < 32"!] [!VAR "temp2"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 8)))))"!][!// [!ELSE!] [!VAR "temp2"="node:value(string(concat('DmaSource',string(num:i((num:i(substring-after(DmaHwChannel,'eDMA_'))+32) div 16)))))"!][!// [!ENDIF!] [!IF "$temp2 = concat('ADC_SAR_',$Unit,'_EOC')"!] [!VAR "foundDma"="1"!] [!VAR "mcl_Id"="string(MclDMAChannelId)"!] [!ENDIF!] [!ENDIF!] [!ENDLOOP!] [!ENDLOOP!] Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The code should checks what channel is enabled before use that channel for processing Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In the Adc_RegOperations.m, replace the below code: [!NOCODE!] [!FOR "Unit" = "0" TO "num:i($maxUnit)"!] [!IF "AdcHwUnit/*/AdcHwUnitId = concat('ADC',$Unit)"!] [!IF "AdcHwUnit/*/AdcHwUnitId[.=concat('ADC',$Unit)]/../AdcTransferType = 'ADC_DMA'"!] [!LOOP "node:refs('ASPathDataOfSchema:/M4_XDM_AR_PKG_NAME/Mcl')/MclConfigSet/*[1]/DMAChannel/*"!] [!VAR

Table continues on the next page...

ID	Subtype	Headline and Description
		"temp"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 16)))))"!][!// [!IF "$temp = concat('ADC',$Unit,'_DMA')"!] [!CODE!] (uint8)[!"MclDMAChannelId"!]U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the DMA channel number for ADC [!"$Unit"!] */ [!ENDCODE!] [!ELSE!] [!CODE!] (uint8)255U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the Interrupt channel number for ADC [!"$Unit"!] */ [!ENDCODE!] [!ENDIF!] [!ENDLOOP!] [!ELSE!] [!CODE!] (uint8)255U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the Interrupt channel number for ADC [!"$Unit"!] */ [!ENDCODE!] [!ENDIF!] [!ELSE!] [!CODE!] (uint8)255U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] /* the Interrupt channel number for ADC [!"$Unit"!] */ [!ENDCODE!] [!ENDIF!] [!ENDFOR!] [!ENDNOCODE!] By: [!FOR "Unit" = "0" TO "num:i($maxUnit)"!][!// [!NOCODE!][!// [!VAR "DmaChannelId"="255"!] [!IF "AdcHwUnit/*/AdcHwUnitId = concat('ADC',$Unit)"!] [!IF "AdcHwUnit/*/AdcHwUnitId[.=concat('ADC',$Unit)]/../AdcTransferType = 'ADC_DMA'"!] [!LOOP "node:refs('ASPathDataOfSchema:/M4_XDM_AR_PKG_NAME/Mcl')/MclConfigSet/*[1]/DMAChannel/*[MclDMAChannelEnable = 'true']"!] [!VAR "temp"="node:value(string(concat('DmaSource',string(num:i(num:i(substring-after(DmaHwChannel,'eDMA_')) div 16)))))"!][!// [!IF "$temp = concat('ADC',$Unit,'_DMA')"!] [!VAR "DmaChannelId"="num:i(MclDMAChannelId)"!] [!ENDIF!] [!ENDLOOP!] [!ENDIF!] [!ENDIF!] [!ENDNOCODE!][!// /** The DMA channel number for ADC [!"$Unit"!] */ (uint8)[!"num:i($DmaChannelId)"!]U[!IF "$Unit < num:i($maxUnit)"!],[!ENDIF!] [!ENDFOR!][!//",'Add a checking in template file if DMA channel is enabled or not
ENGR00366489	Defect	[ADC] The function Adc_ReadGroup return E_NOT_OK but it also changes state of the group that is its argument to ADC_IDLE

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): - Without interrupt is enabled. - Group0 is configured as one-shot, sw trigger (limit checking is optional) - Procedure: 1. Adc_Init() 2. Adc_SetupResultBuffrer(); 3. Adc_StartGroupConversion(Group0); 4. Ret = Adc_ReadGroup(Group0); 5. EU_ASSERT(E_NOT_OK==Ret); 6. /* Verify the state of Group 0 is ADC_BUSY */ EU_ASSERT(ADC_BUSY==Adc_GetGroupStatus(Group0); The problem here is that the state of Group 0 is ADC_IDLE. Between step 3 and step 4. The function Adc_ReadGroup must be called such that there is no available result (hardware does not finish conversion). The problem also appears on RR. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The driver should not raise an error at the configuration time when configure a software trigger group with Adc_StartStopGroup API is switched off. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): n the IPV_ReadGroup function, we have below output cases: 1. Without interrupt == ON - Return = E_NOT_OK and bFlag = TRUE: Conversion is not yet finished, no need to update group status. - Return = E_OK and bFlag = TRUE: Conversion is finished, limit checking is passed, need to update group status. - Return = E_NOT_OK and bFlag = FALSE: Conversion is finished, limit checking is falied, need to update group status. 2. Without interrupt == OFF - Return = E_NOT_OK and bFlag = TRUE: Conversion is not yet finished or limit checking is falied, no need to update group status. - Return = E_OK and bFlag = TRUE: Conversion is finished, need to update group status. In both Without interrupt == ON and OFF, if we have (Return == E_NOT_OK and bFlag == TRUE), we don't need to update group status. So, in the Adc_ReadGroup function, the code should be: GroupRet = Adc_Ipw_ReadGroup(Group, pDataBufferPtr, &bFlag); /*if the conversion is finished or if the limit checking was failed for the group without interrupts*/ if((GroupRet == E_OK) || (bFlag == FALSE)) { /* Update group status */ Adc_UpdateStatusAfterReadGroup(Group, bFlag); }","The function Adc_ReadGroup does not update group status if conversion if not yet completed or limit checking is failed in case group without interrupt is OFF

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00377955	NewWork	[ADC] The function Adc_SetClockMode and Adc_Calibrate should ensure there is no conversion ongoing before process their functionality 'NewWork Description: Currently, the function Adc_SetClockMode and Adc_Calibrate don't check Adc unit is busy or not before process their functionality. This might cause Adc stuck if changing clock mode or process calibration during Adc is in conversion. Requirement source: cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): If development error detection for the ADC module is enabled: if called when ADC hardware status is BUSY or if the group conversion is on-going, the function Adc_SetClockMode/Adc_Calibrate shall raise development error ADC_E_BUSY and return without any action.",'Added checks for ongoing conversions for APIs Adc_SetClockMode and Adc_Calibrate
ENGR00378476	NewWork	[ADC] The function Adc_SetClockMode will not be depended on AdcConvTimeOnce 'NewWork Description: Currently, Adc_SetClockMode is enabled if AdcConvTimeOnce is enabled only. This will forces user uses AdcConvTimeOnce conversion time once if want to use Adc_SetClockMode. Actually, Adc_SetClockMode should not be depended on AdcConvTimeOnce. The driver will sets conversion time if AdcConvTimeOnce is enabled, if not, Adc_SetClockMode will changes clock only. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Adm.xdm: Remove the Invalid of node AdcEnableDualClockMode - Adc_Adcdig.c: guard the code that sets conversion time by #if (ADC_SET_ADC_CONV_TIME_ONCE == STD_ON) in function Adc_Adcdig_SetClockMode",'The function Adc_SetClockMode will not be depended on AdcConvTimeOnce,ENGR00381023
ENGR00380221	Defect	[ADC] There are two INVALID attribute in AdcGroupDefinition node 'Problem detailed description (how to reproduce it): There are two separate INVALID attribute in node AdcGroupDefinition in Adc.xdm: It should be merged into one INVALID attribute. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: The INVALID attribute should be updated as bellow: <a:da name="INVALID" type="XPath"> <a:tst expr=".='" true="Select an ADC channel for the configured group."/> <a:tst expr="contains(., ../../../../@name)" false="The ADC channel must be mapped on the same Hw Unit Group."/> </a:da> Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update INVALID attribute as bellow: <a:da name="INVALID" type="XPath"> <a:tst expr=".='" true="Select an ADC channel for the configured group."/> <a:tst expr="contains(., ../../../../@name)" false="The ADC channel must be mapped on the same Hw Unit Group."/> </a:da>','Update invalid of node AdcGroupDefinition
ENGR00372078	Defect	[ADC] There is no reporting error from Adc_DeInit while conversion is ongoing 'Problem detailed description (how to reproduce it): As description in the requirement ADC112 the function Adc_DeInit will raise ADC_E_BUSY when the conversion is ongoing (for group that can not stop implicitly and in state ADC_STREAM_COMPLETED). However, currently there is no raising error. Procedure: Group 8: Hw trigger, single access, - Adc_EnableHardwareTrigger (AdcGroup_8); - Start trigger to ADC till the state of Group 8 is ADC_STREAM_COMPLETED. - Invoke function Adc_DeInit() - Expect error: ADC_E_BUSY is raised. At this step, there is no error. Preconditions: - The group 8 is configured as hw trigger, single access, not limit number of channels in the group. Test Case ID (internal TC that caught the defect) - optional Adc_TC_0070 Trigger: - The function Adc_DeInit is invoked while Group 8 is in state ADC_BUSY Observed behavior: - After invoke Adc_DeInit. The DET error is not raised Expected behavior: The Det error (ADC_E_BUSY) is raised Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]','ADC_E_BUSY will be raised if it has any group in queue.
ENGR00364407	NewWork	[ADC] Update ADCDIG to support Treerunner 'NewWork Description: Update IPV_ADCDIG to support Treerunner','Add code to support TreeRunner platform
ENGR00370592	NewWork	[ADC] Update Base address array to display all HW Units

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: N/A Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the adress 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....",'Update the base Address array to display all the HW Units
ENGR00380322	Defect	[ADC] Update the ADC with the change for defines in MCL 'Problem detailed description (how to reproduce it): To fix the issue in the ticket ENGR00379359, IPV_DMA of MCL re-defined some Macros for Flags of DMA. From #define DMA_TCD_DONE_U32 #define DMA_TCD_ACTIVE_U32 #define DMA_TCD_MAJOR_E_LINK_U32 #define DMA_TCD_E_SG_U32 #define DMA_TCD_DISABLE_REQ_U32

Table continues on the next page...

ID	Subtype	Headline and Description
		#define DMA_TCD_INT_HALF_U32 #define DMA_TCD_INT_MAJOR_U32 #define DMA_TCD_START_U32 To #define DMA_TCD_DONE_U8 ((uint8)0x80U) #define DMA_TCD_DONE_U32 ((uint32)DMA_TCD_DONE_U8) #define DMA_TCD_ACTIVE_U8 ((uint8)0x40U) #define DMA_TCD_ACTIVE_U32 ((uint32)DMA_TCD_ACTIVE_U8) #define DMA_TCD_MAJOR_E_LINK_U8 ((uint8)0x20U) #define DMA_TCD_MAJOR_E_LINK_U32 ((uint32)DMA_TCD_MAJOR_E_LINK_U8) #define DMA_TCD_E_SG_U8 ((uint8)0x10U) #define DMA_TCD_E_SG_U32 ((uint32)DMA_TCD_E_SG_U8) #define DMA_TCD_DISABLE_REQ_U8 ((uint8)0x08U) #define DMA_TCD_DISABLE_REQ_U32 ((uint32)DMA_TCD_DISABLE_REQ_U8) #define DMA_TCD_INT_HALF_U8 ((uint8)0x04U) #define DMA_TCD_INT_HALF_U32 ((uint32)DMA_TCD_INT_HALF_U8) #define DMA_TCD_INT_MAJOR_U8 ((uint8)0x02U) #define DMA_TCD_INT_MAJOR_U32 ((uint32)DMA_TCD_INT_MAJOR_U8) #define DMA_TCD_START_U8 ((uint8)0x01U) #define DMA_TCD_START_U32 ((uint32)DMA_TCD_START_U8) So, instead of use the define with _U32, the ADC should use the defines with _U8 (ex : DMA_TCD_MAJOR_E_LINK_U8 to fix the Misra Preconditions: Mcl_DmaTcdSetFlags function is used. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Misra 10.1 will be raised in MCL users. Expected behavior: No MISRA error will be raised in MCL users becasue of MCL. 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",'MISRA error reported in MCL users when Mcl_DmaTcdSetFlags function is used.
ENGR00380832	NewWork	[ADC] Update use of Mcl_DmaTcdAttributesType according to Mcl updates 'NewWork Description: There are some elements of struct in Mcl_Dma_Types.h that are renamed in the latest version of Mcl driver. IPV_ADCDIGV2 files use this data type, so the code needs to be updated accordingly. Below is more details: typedef struct { VAR(uint32, AUTOMATIC) saddr; /** @brief source address */ VAR(uint32, AUTOMATIC) daddr; /** @brief destination address */ VAR(uint32, AUTOMATIC) ssize; /** @brief source transfer size */ VAR(uint32, AUTOMATIC) dsize; /** @brief destination transfer size */ VAR(uint32, AUTOMATIC) soff; /** @brief source address offset */ VAR(uint32, AUTOMATIC) doff; /** @brief destination address offset */ VAR(uint32, AUTOMATIC) smod; /** @brief source address modulo */

Table continues on the next page...

ID	Subtype	Headline and Description
		VAR(uint32, AUTOMATIC) dmod; /** @brief destination address modulo */ VAR(uint32, AUTOMATIC) num_bytes; /** @brief number of bytes to be transferred */ VAR(uint32, AUTOMATIC) iter; /** @brief iteration count */ }Mcl_DmaTcdAttributesType; to: typedef struct { VAR(uint32, AUTOMATIC) u32saddr; /** @brief source address */ VAR(uint32, AUTOMATIC) u32daddr; /** @brief destination address */ VAR(uint32, AUTOMATIC) u32ssize; /** @brief source transfer size */ VAR(uint32, AUTOMATIC) u32dsize; /** @brief destination transfer size */ VAR(uint32, AUTOMATIC) u32soff; /** @brief source address offset */ VAR(uint32, AUTOMATIC) u32doff; /** @brief destination address offset */ VAR(uint32, AUTOMATIC) u32smod; /** @brief source address modulo */ VAR(uint32, AUTOMATIC) u32dmod; /** @brief destination address modulo */ VAR(uint32, AUTOMATIC) u32num_bytes; /** @brief number of bytes to be transferred */ VAR(uint32, AUTOMATIC) u32iter; /** @brief iteration count */ }Mcl_DmaTcdAttributesType; So the reference in IPV_ADCDIGV2 should be updated Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Mcl_DmaTcdAttributesType is change from",'Update the reference when Mcl_DmaTcdAttributesType was changed,ENGR00380846
ENGR00373138	NewWork	[ADC] Updates for compatibility between ASR4.0 and ASR4.2 implementations 'NewWork Description: Updates for compatibility between ASR4.0 and ASR4.2 implementations. - Limit checking parameters are separated in Adc_Adcdig_ChannelLimitCheckingType structure - Limit checking configuration should be identical in all configuration sets, becasue these parameters are defined as PreCompile in ASR4.0.3 specification. Adc.xdm must be enhanced with a check for this. - Registers availability information is separatred in Adc_Adcdig_RegisterAvailabilityType - add definition for ADC_CTUV2_UNIT1_AVAILABLE on CTU2 platforms (jsut because of porting on RR ultra) - Review Adc.xdm according to ASR4.0.3 parameter specification. for example, multiplicity 0..1 is not fulfilled by AdcClockSource, so optional=true should be added. this is not checked by the current vsmd tool. - add AdcLogicalUnitId parameter. - Adc.xdm: remove AdcGroupAction, AdcSetOnceRegisters, AdcEnableSelfTest, Safety container - Adc.xdm: move AdcPriorityQueueMaxDepth to AdcGeneral and remove AdcGeneric container. Expected behavior: N/A Requirement source: N/A (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): N/A","- Limit checking parameters are separated in Adc_Adcdig_ChannelLimitCheckingType structure - Registers availability information is separatred in Adc_Adcdig_RegisterAvailabilityType
ENGR00373390	NewWork	[ADC] Use "READONLY" instead of "EDITABLE" in xdm file 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1) Nodes that have constant false editable conditions (<a:a name="EDITABLE" value="false"/>) will change to (<a:a name="READONLY" value="true"/>) 2) Exception: the 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/> 3) Nodes that have EDITABLE attributes with type xPath will not be changed to READONLY",'Replaced EDITABLE with READONLY in xdm confiugration file.
ENGR00375505	Defect	[ADC] Wrong EDITABLE expression for AdcGroupPriority parameter 'Problem detailed description (how to reproduce it): The EDITABLE expression for AdcGroupPriority is referring to wrong path of AdcPriorityImplementation. Current path: ../../../../../../AdcGeneral/AdcPriorityImplementation. Correct path: ../../../../../../../AdcGeneral/AdcPriorityImplementation In addition, some optional parameter is referred but don't check existing. Example for: AdcChannelLimitCheck. Please review all the path of all expressions and optional parameters and correct if any. 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 Trigger: N/A Observed behavior: N/A Expected behavior: Correct all of path 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",'Correct path of AdcPriorityImplementation
ENGR00372066	Defect	 [ADC] Wrong comment for misra error 'Problem detailed description (how to reproduce it): Some comment for misra is wrong: In the Adc_PBcfg.c, the comment below should be corrected: /** * @violates @ref Adc_Cfg_c_REF_1 only preprocessor statements and comments before "#include" * @violates @ref Adc_Cfg_c_REF_2 Repeated include file Memmap.h */ Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The comment for misra should be corrected as below: /** * @violates @ref Adc_PBcfg_c_REF_1 only preprocessor statements and comments before "#include" * @violates @ref Adc_PBcfg_c_REF_2 Repeated include file Memmap.h */ 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",'Fix missra error from Adc_Cfg.c and Adc_PBcfg.c,ENGR00372436
ENGR00368685	Defect	 [ADC] unused Adc_UnitStatusType structure members 'Problem detailed description (how to reproduce it): CurrentChannel and bInjConvOngoing are set but not referenced in MCAL Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		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",'Remove the unused element from Adc_UnitStatusType: CurrentChannel and bInjConvOngoing,ENGR00368979
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" #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",'Added pragma for generating far call for flash RAM function.
ENGR00367327	NewWork	[BASE] Add IP versions for IPV_FXOSC 'NewWork Description: Add IP versions for IPV_FXOSC on Calypso Expected behavior: IPV_FXOSC should have a version Requirement source: none (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Add IP versions for IPV_FXOSC on Calypso
ENGR00377850	NewWork	[BASE] Add eMios errata ERR009978 in BASE for Calypso 'Problem detailed description (how to reproduce it): Errata ERR009978 must be added in base in Base for Calypso 3M and 6M: Changing mode from GPI to MCB can cause a spurious interrupt Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		eMios Ch8 is used to generate Pwm signal, and Ch14 is used in the Gpt driver as one shot mode, with an internal counter. Test Case ID (internal TC that caught the defect) - optional NA Trigger: Having Ch8 as Pwm generator Observed behavior: Interrupt sometime occured immediately after changing to MCB mode (when ussing Gpt_StartTimer) 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): Follow the following steps in Gpt_eMios_StartTimer: - Save interrupt enable bit - Clear interrupt enable bit - Change the mode from GPI to MCB - Clear the interrupt flag - Write the interrupt enable bit, with the saved value Requirement source: none (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Fixed hardware issue with spurious interrupts when chaning from GPI to MCB mode.
ENGR00366828	NewWork	[BASE] Add support for I2C driver 'NewWork Description: Add support for I2C driver in Base Expected behavior: Add support for I2C - memmapt sections, base address, memory and pointer classes Requirement source: Customer Request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Added support for I2C 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)",'Add support for different core architectures
ENGR00380977	NewWork	[BASE] Align the naming convention for the XBAR define, in Reg_eSys.h file

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: Right now, the defines for the BASEADDR for XBAR are different between platforms.: E.g.: On Cobra55: #define XBAR_BASEADDR ((uint32)0xFFF04000UL) On Calypso: #define XBAR0_BASEADDR ((uint32)0xFC004000UL) Requirement source: none (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): All platforms should have the same naming convention: #define XBAR_0_BASEADDR","'All platforms should have the same naming convention: #define XBAR_0_BASEADDR
ENGR00372198	NewWork	[BASE] Implement findings after code review against checklist for RTM release 'Please correct the problems found during the code review against checklist. The list of issues is reported in the Remarks column of the Excel report attached to this CR. Please consider all rows having the Status column set to value 'Failed'",'Implement code review with checklist findings
ENGR00369066	NewWork	[BASE] Move the Java files from Gpt to Base '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=..",'Add JAR file in Base
ENGR00373391	NewWork	[BASE] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true",

Table continues on the next page...

ID	Subtype	Headline and Description
		the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.','replace EDITABLE with READONLY for VendorApiInfix variable
ENGR00370098	NewWork	[CAN] Activate Elektrobit Tresos Fibex import '# Enable/Disable comimporter information within generated plugin.xml file ## PLG_USE_COMIMPORTER = false PLG_USE_COMIMPORTER should be true','Can.xdm
ENGR00373741	NewWork	[CAN] Add support for selecting between ISO and non-ISO CAN FD 'NewWork Description: In the old hardware, there is only one option to use ISO or non-ISO CAN FD depend on the IP version. On TreeRunner and RaceRunner Ultra, there is is a bit (bit 19: STFCNTEN) in CAN_CTRL2 to select between ISO CAN FD and non-ISO CAN FD. Therefore it is possible to let users select whether they want to use ISO or non-ISO CAN FD to compatible with their device. Proposed solution (Optional): Add a check box in xdm to configure ISO or non-ISO CAN FD.","'Can.xdm Can_Cfg.h Can_PBCfg_42.c",ENGR00375722
ENGR00361467	Defect	[CAN] Avoid multiple definition of STATIC macro 'Problem detailed description (how to reproduce it): STATIC macro should be removed from Mcal.h; this definition is not required by Autorsar 4.0 standard. All components should use the C keyword 'static' instead. 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 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): NA','Removed definition of STATIC macro.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00368891	Defect	[CAN] Can driver check max_mb incorrectly 'Problem detailed description (how to reproduce it): + When CanControllerFdBaudrateConfig is disabled: Can driver need check max_mb for every controller follow max_mb of every controller, ex: In Kinetis platform FlexCAN_A has 32 MBs, FlexCAN_B has 16MBs, FlexCAN_C has 16MBs. But Can driver still check max_mb follow max_mb of FlexCAN_A (32 MBs) + When CanControllerFdBaudrateConfig is enabled: Can driver need check max_mb follow max_mb of every controller and Payload size, ex: In Kinetis platform with FLexCAN_A: Payload Max_MB 8 32 16 21 32 12 64 7 But Can driver still check max_mb follow max_mb of FlexCAN_A (32 MBs) 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): [...]",'Can.c
ENGR00370228	Defect	[CAN] CanAutoBusOffRecovery description in EB plugin and Note in UM is incorrect 'Problem detailed description (how to reproduce it): CanAutoBusOffRecovery description in EB plugin and Note in UM is incorrect With CanAutoBusOffRecovery checked, the following code is generated. /* CTRL[BOFF_REC] - Bus off recovery */ ((uint32)0 << FLEXCAN_CTRL_BOFFREC_SHIFT_U8) | And with CanAutoBusOffRecovery unchecked, the following code is generated. /* CTRL[BOFF_REC] - Bus off recovery */ ((uint32)1 << FLEXCAN_CTRL_BOFFREC_SHIFT_U8) | This does mean that the CanAutoBusOffRecovery checked for enabled Automatic Bus Off Recovery and unchecked to disable this feature. However the EB Plugin description and UM Note says: Enable/Disable automatic BusOff recovery (CTRL[BOFF_REC] bit). 0(Unchecked) = Automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. 1(Checked) = Automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated(zero) by the user. Proposed solution (Optional): Correct UM Note and the description in EB Tresos Plugin to: Enable/Disable automatic BusOff recovery (CTRL[BOFF_REC] bit). 0(checked) = Automatic recovering from Bus Off state occurs according to the

Table continues on the next page...

ID	Subtype	Headline and Description
		CAN Specification 2.0B. 1(Unchecked) = Automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated(zero) by the user.",'Can.xdm
ENGR00365378	Defect	[CAN] Compiler Warning is not commented 'Problem detailed description (how to reproduce it): Some warnings are not commented. Please see attached file in Attachments tab. 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): Move some variables into Can FD checking.",'Fix compiler warning in Can_Flexcan.c
ENGR00374648	Defect	[CAN] Correct Misra 'Problem detailed description: When run test, the tester found some misra error. Expected behavior: Don't exist any misra error. Proposed solution (Optional): Check the log file and fix all misra error.",'update Can.c and Can_IPW.h,ENGR00374650
ENGR00381457	Defect	[CAN] Correct misra violation 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate both rule 11.1 and 11.3, only rule 11.1 is reported. When issue in tool is fixed, we have to fix or comment the violations related to these rules. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Have error messages prevent generating Misra report Expected behavior: Can generate Misra report without any error

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): Fix the violation if possible or comment the violations with deviation reasons",'Fix misra violation,ENGR00381462
ENGR00379733	Defect	[CAN] Disable CanSetBaudrateApi for all platform do not support can FD 'Problem detailed description (how to reproduce it): This feature is for platform which support FD, to switch between different baudrate. However, without FD, this function does not have much meaning. It was mistake enable for all platforms which cause confusing for platform. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: This feature is disable for platform which does not support FD Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Disable CanSetBaudrateAPI for platform does not support FD.
ENGR00355144	Defect	[CAN] Fix Misra error 'Problem detailed description (how to reproduce it): There is new misra error due to change of DET. This CR will fix this issue. Preconditions: N/A 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",'Fixed Misra violation
ENGR00372759	Defect	[CAN] Fix Misra error 'Problem detailed description (how to reproduce it): In the IPV_FlexCan.c file have some Misra error. Expected behavior: All Misra error is fixed. Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		Codding guideline Propose solution: Analyze and fix all Misra error.",'Can_FlexCan.c
ENGR00374756	Defect	[CAN] Fix VSMD issue report from Halo release. 'Problem detailed description: In the Can.xdm file has four parameter which have the "INTEGER" type, but don't have the check range for it. This problem made the VSMD issue. Expected behavior: This issue don't exist in the VSMD report. Proposed solution (Optional): separate the check "<a:da name="INVALID" type="XPath">" to two check following: <a:da name="RANGE" type="XPath">: check range of value. <a:da name="RANGE" type="XPath">: check the on/off of the "CanAdvancedSetting".",'Can.xdm
ENGR00376014	Defect	[CAN] Fix build error when build by IAR 'Problem detailed description: One build error happen when build code by IAR compiler, this error related the section memory where put the "CanStatic_ConfigSet" structure. Expected behavior: build successful by IAR compiler.",'Can_Cfg.c
ENGR00374664	Defect	[CAN] Fix compiler warning for Halo release 'Problem detailed description: Some warnings are not commented. Please see attached file in Attachments tab. Expected behavior: All compiler warning is fixed. Requirement source: Codding guideline Propose solution: Analyze and fix all compiler warning.",'Can_FlexCan.c,ENGR00374671
ENGR00373294	Defect	[CAN] Fix compiler warning related ControlerDescriptors<x>_PB when controller is inactivated 'Problem detailed description: When configure more than 1 controllers and have one which is not activated. In this case, the Baudrate config for that controller is not reference because the inactivated controller is not able to use. However, this cause compiler warning because we did generate the Baudrate config for inactivated controller. This is not a functional issue, but to remove this warning, we need to update the implementation of generate template for inactivated controller. Root cause: If in the configuration, the user disable one any controller. So all information of this controller will be generated as the latest element of "ControlerDescriptors" structure with the default value.

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: All the compiler warning is resloved. Requirement source: Codding guideline Propose solution: The driver should be generate all information of each controller into "ControlerDescriptors" tructure without don't need to check the controller activation. The Can driver will use the NULL_POINTER in order to detect the controller which is disabled.','Does not generate baudrate config for inactivated controller
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 Proposed solution: It need check the u8RxFifoTableId before write",'Update Can_FlexCan_ChangeBaudrate in Can_Flexcan.c
ENGR00363040	Defect	[CAN] ISRs does not check for driver initialization according to FMEA requirements 'Problem detailed description (how to reproduce it): Source code is not full cover FMEA requirements as the following requirements ask: [SMCAL_SW035] ISR shall check whether its respective driver is initialized. If the driver is not initialized, the ISR shall only clear interrupt status flag and return immediately. CAN driver ISRs does not check that the driver is in CAN_INIT state as the FMEA require. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		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 interrupt status flags shall be cleared even if the driver was not initialized. The interrupt status flags shall be checked and then cleared if needed(flags are set) inside ISR even some of them(Tx, Rx) are cleared also inside complex processing functions(Can_FlexCan_ProcessTx, Can_FlexCan_ProcessRx). This approach will allow us to have an easy to maintain source code even some interrupt status flag checks are made twice. e.g. Can_FlexCan_RxFifoFrameAvNotif need check CAN_INIT state before run its function.","'Update Can_Flexcan.c, Can_Flexcan.h
ENGR00378344	NewWork	[CAN] 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","'Review, analysis and implement Exclusive area for all global variables and registers which need to protect
ENGR00381559	Defect	[CAN] Implement errata e10368 'Problem detailed description (how to reproduce it): The activation or deactivation of the CAN FD operation by setting or clearing the FDEN bit of the CAN_MCR register or by setting the FlexCAN soft reset bit (SOFTRST) of the CAN_MCR register when the FDEN bit is enabled may cause an internal FlexCAN register to become metastable. As result, the first CAN frame, transmitted or received, may have corrupted data (ID and payload). However, even though the data is corrupted, a valid CAN frame is transmitted because the Cyclic Redundancy Check (CRC) calculation is based on the corrupted data. During reception the data is corrupted internally after the CRC bits have been checked and therefore this corrupted data may be stored in a reception message buffer. After the first CAN frame, all subsequent frames are transmitted and received correctly. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Transition of the CAN FD operation enable bit may lead FlexCAN logic to an inconsistent state Expected behavior: Perform the steps in Proposed solution to set the FDEN bit Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Perform the following steps to set the FDEN bit: 1. If FlexCAN is already in freeze mode, go to step 3, otherwise set the HALT and FRZ bits of the CAN_MCR register. 2. Wait the FRZACK bit of the CAN_MCR register to be set by the hardware. 3. Set the LPB (Loop Back Mode) bit of the CAN_CTRL1 register.

Table continues on the next page...

ID	Subtype	Headline and Description
		4. Configure only one message buffer to be transmitted. The frame should be a classical one (non-FD) with IDE =0, RTR =1 DLC =0x5 and STD_ID =0x682. 5. Set the FDEN bit of the CAN_MCR register. 6. Clear the HALT bit of the MCR register to leave freeze mode. 7. Wait the FRZACK bit of the CAN_MCR register to be cleared by the hardware. 8. Wait the respective bit of the CAN_IFLAG register to be set (successfully transmission in loop back mode). 9. Clear the respective bit of the CAN_IFLAG register by writing 1. 10. Set the HALT and FRZ bits of the CAN_MCR register. 11. Wait the FRZACK bit of the CAN_MCR register to be set by the hardware. 12. Clear the LPB (Loop Back Mode) bit of the CAN_CTRL1 register.",'Implement errata e10368 for enabling the FDEN bit
ENGR00371963	Defect	[CAN] Implementation of errata e9527 'Problem detailed description (how to reproduce it): The Flexible Controller Area NEtwork (FlexCAN) is not able to abort a transmission frame and the abort process may remain pending in the following cases: a) If a pending abort request occurs while the FlexCAN is receiving a remote frame. b) When a frame is aborted during an overload frame after a frame reception. c) When an abort is requested while the FlexCAN has just started a transmission. d) When Freeze Mode request occurs and the FlexCAN has just started a transmission 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): Implement errata workaround",'update Can_Flexcan.c
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.",'Update UM to have more detail about DET errors

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00377829	Defect	[CAN] Improve code for the user mode. 'Problem detailed description (how to reproduce it): When implement for user mode, developer added one element in the can.xdm file in order to enable/disable this feature. But the default value of this element is "true". This problem will make the build error for other platform which don't support user mode. Expected behavior: The source code which is implemented for user mode must compatible with all platform. Even platform does not support user mode.","'Can.mk Can.xdm
ENGR00372073	NewWork	[CAN] Improve code indentation and add more description comments in the generated code templates 'NewWork Description: Current, in the Can driver exist some tab characters. Also, in the generation code and the driver code, the code indentation is not good. Expected behavior: Improve code indentation and add more description comments in the generated code templates.Remove also the empty lines. Also, if needed the indentation shall be improved into other driver files. Also tab characters shall be removed. Requirement source: Codding guideline Proposed solution (Optional): Check in all the generation code and driver code in order to correct code indentation and remove the tab characters.','update Can_Mcan.c
ENGR00377684	Defect	[CAN] Improve code trigger transmission 'Problem detailed description: This CR is a small issue generated when implement TriggerTransmit (a feature in ASR 4.2.1). During implement this feature, the developer made the command eReturnValue = E_OK work 2 times (it is because the maintain similarities between ASR 4.2 and ASR 4.0.3 in IPV file). And therefore it is not code optimized version even it does not impact to functionality. About the config affected, it would impact only platform release after Kinetis Beta 0.9.0 ASR 4.2. This CR impacts IPV, then it will affect to release which use baseline BLN_IPV_FLEXCAN_SMCAL_4.0_01.34.00 and above. Preconditions: NA Expected behavior: Execution of define value for eReturnValue should be done only once.','Can_FlexCan.c
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:

Table continues on the next page...

ID	Subtype	Headline and Description
		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, #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). * */

Table continues on the next page...

ID	Subtype	Headline and Description
		#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):",'Can.h
ENGR00370151	Defect	[CAN] Improve naming conventions according to coding guideline 'Problem detailed description (how to reproduce it): In the Reg_eSys_FlexCan.h file, exist the naming convention incorrectly at the line 591 to line 598. the definition line don't have the "FLEXCAN" in the name of definition. Expected behavior: The definition line should be have the "FLEXCAN" at beginning point Requirement source: ASR Propose solution: Correct the naming convention.","'Reg_eSys_FlexCan.h Can_Flexcan.c",ENGR00372089
ENGR00368815	NewWork	[CAN] Improve the implementation of parameters in CanControllerBaudrateConfig 'NewWork Description: 1. In the current implementation, some parameters in CanControllerBaudrateConfig (CanControllerPropSeg, CanControllerSeg1, CanControllerSeg2, CanControllerSyncJumpWidth) are defined in configuration as values which will be configure in HW registers. Therefore, the configure values are different based on IP_Vaults (MCAN and FlexCan). Users have to counter this value when configure for specific HW. For example, in FlexCAN based platform we need to add 1 for each; for MCAN we also added one to each of them except for ProgSeg. And also it was different for FlexCAN based HW when configure for FD and normal Baud rate. Additionally, ASR define the unit for them as time-quanta. However, current implementation need to add extra value for them. This could be consider as not fully implemented for corresponded ASR requirements. 2. Suggest to move the bit timing related range checks in Can_PBcfg.c into the Can.xdm so that the user immediately sees the problem without having to launch the code generator.

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: The values configured should be used for calculating the time value without extra. Requirement source: (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The driver will base on configured value and calculate proper value for register.","'update Can.xdm update Can_Cfg.c update Can_PBCfg.c
ENGR00380670	NewWork	 [CAN] Improve xdm configuration related to baudrate 'NewWork Description: There are the following improvements could be applied for xdm file: - There isn't configured value for SYN_Segment event the value for this node is always 1 and is counter when calculate the baud rate and time quanta. This might cause confusion for user when they don't know why it is required to add 1 to the rest. So please create the node configuration to EB for inform that the time quanta also included Syn segment with default value =1. - The description for CanSoftwareBusOffRecovery is not very clear. This might confuse user on how to configure this parameter. - The CanControllerCBT container and its values should be enabled and editable independently with the RX FIFO and the FD features. We should define a parameter in resource file to enable the CanControllerCBT container for platform which has CAN_CBT register. - Don't necessary to use #if(condition) for the nocode-statements (in Can_cfg.c and Can_PBcfg.c) which aren't generated. For example #if (CAN_FD_MODE_ENABLE == STD_ON) in the ControllerBaudrateConfigs_PC_[!"CanControllerId"!] struct: [!/* Check the CAN FD bit time compliance - get the values from Tresos interface and check to see that the number of time quantas introduced by user it is correct.*/!] [!/* Formula used fot checking the total number of time quantas is: NoTqFD= fcanclk/((desired_baudrate[Hz])*(FD_prescaller +1))*/!] #if (CAN_FD_MODE_ENABLE == STD_ON) [!IF "(node:exists(CanControllerFdBaudrateConfig)) = 'true'"!] [!SELECT "node:ref(../../CanCpuClockRef)"!] [!VAR "OutCanClockFrequencyFromMcuFd" = "McuClockReferencePointFrequency"!] [!ENDSELECT!] - CanTimeoutDuration is define as number of loop cycle, not map with configuration file: <a:da name="INVALID" type="Range"> <a:tst expr=">=0.001"/> <a:tst expr="<=65.535"/> </a:da> Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Add extra node into EB to describe Syn_Segment. - Modify the label of CanSoftwareBusOffRecovery to "Can Enable Software BusOff Recovery".

Table continues on the next page...

ID	Subtype	Headline and Description
		- Modify the CanControllerCBT container according to the above description. - Remove unnecessary #if(condition) statements in the body of nocode-statement. - Correct the range of CanTimeoutDuration in configuration file.','Improve xdm configuration related to baudrate,ENGR00380845
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. - 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];","'Can.c Can.h
ENGR00373410	Defect	[CAN] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) '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. For CAN driver, variable CanStatic_ConfigSet was extern in file Can.h in section SEC_CODE which is not identical with section SEC_CONFIG_DATA_UNSPECIFIED where define this variable. 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): Move variable CanStatic_ConfigSet from SEC_CODE to SEC_CONFIG_DATA_UNSPECIFIED.",'Update to use same memap section for all declarations of each variable
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. Preconditions: Using FIFO in polling mode Test Case ID (internal TC that caught the defect) - optional NA Trigger: Overflow occur Observed behavior: Does not able to clear the overflow state Expected behavior: Process all message and able to exit overflow state.

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",'Correct the behavior when fifo overflow and fifo warning occurred.
ENGR00365666	Defect	[CAN] 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 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","'Update Can_VersionCheck_Inc.m Can_VersionCheck_Src.m Can_VersionCheck_Src_PB.m
ENGR00373905	NewWork	[CAN] Investigate and implemen to reduce cyclomatic complexity and nesting level 'NewWork Description: - 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. 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 Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): In AUTOSAR_MCAL_<MDL>_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments

Table continues on the next page...

ID	Subtype	Headline and Description
		- for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth >4 the problems shall be resolved",'Reduced cyclomatic complexity and nesting level
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. 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",'can.xdm
ENGR00363365	Defect	[CAN] MB address is calculated incorrect 'Problem detailed description (how to reproduce it): MB address is calculated incorrect when there are more than one controller configured. First configured with FD and second without FD 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): [...]",'Can_FlexCan.c and Can_FlexCan.h

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00379720	NewWork	[CAN] Perform code review against checklist for Calypso 3M/6M RTM 1.0.2 'NewWork Description: Review code review against checklist Implement findings Update Release note field to "Y" if you change anything in code/UM/IM according to this review. Otherwise, leave it as "N". Requirement source: sMCAL Release criteria document: http://compass.freescale.net/go/228798570 Coding guideline: http://compass.freescale.net/go/227962899 Code review checklist: http://compass.freescale.net/go/224108405 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Updated naming for some variable + macro,ENGR00380579
ENGR00376211	NewWork	[CAN] Remove the code related to wakeup on bus activity feature 'NewWork Description: According to the attached discussion, the hardware team informed us that the wake up on bus activity feature is not supported by the existing platforms which includes Flexcan controllers. Also, according to HW RM, this feature is not supported neither by MCAN controllers. Based on that info, it was decided to remove from CAN driver all existing code related to wake up on bus activity feature because it is not needed and to improve the code readability and maintainability.","'Can.xdm Can_Cfg.h Can_Cfg.c Can_PbCfg.c Can.h Can_IPW.h Can.c Can_Irq.c",ENGR00377727
ENGR00372710	Defect	[CAN] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. Expected behavior: All comments that start with (c) Copyright M4_<>_YEAR_ID shall have an M4 tag for the copyright to M4_<>_COPYRIGHTED_TO Proposed solution (Optional): Check all driver and correct follow: In the generation file: - replace the "Freescale Semiconductor Inc. & STMicroelectronics" by "M4_PLG_COPYRIGHTED_TO". - replace the "M4_SRC_COPYRIGHTED_TO" by "M4_PLG_COPYRIGHTED_TO".","'- The geration file: Can_VersionCheck_Inc.m; Can_VersionCheck_Src.m; Can_VersionCheck_Src_PB.m; Can_NotifyCheck_Src.m; Can_NotifyCheck_Src_PB.m; Can_NotifyCheck_Src.m;

Table continues on the next page...

ID	Subtype	Headline and Description
		Can_NotifyCheck_Src_PB.m. Can_Cfg.h; Can_Cfg.c; Can_PbCfg.c
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","'Support Can_SetBaudrate API (refer to ASR 4.2.1 for detail). Modify in files: Can.xdm, Can.c, Can.h, Can_Cfg.h
ENGR00363154	Defect	[CAN] The function Can_AbortMb raise DET error with wrong function id 'Problem detailed description (how to reproduce it): The function Can_AbortMb raise DET error with wrong function id when call the function with invalid HTH parameter Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: Execution Observed behavior: The function raise DET error with function id: CAN_SID_WRITE Expected behavior: The function raise DET error with function id: CAN_SID_ABORT_MB 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_ReportError in Can_AbortMb in Can.c
ENGR00365372	Defect	[CAN] The value of MBDSRx bits in CAN_FDCTRL register was not reset before setting 'Problem detailed description (how to reproduce it): Configure 2 CFG_SETS: + CFG_SET_1: configure with MBDSR is CAN_64_BYTES_PAYLOAD + CFG_SET_2: configure with MBDSR is CAN_16_BYTES_PAYLOAD Execute test with CFG_SET_1: Transmit and receive are OK Then switch to CFG_SET_2

Table continues on the next page...

ID	Subtype	Headline and Description
		Execute test with CFG_SET_2: Transmit is not OK, Receive is OK Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional tc_fnc_can_00502 in tse_bbx_fnc_can_50200 Trigger: Execution Observed behavior: Transmit is not OK, Receive is OK when switch between 2 CFG_SETS Expected behavior: Transmit is OK, Receive is OK when switch between 2 CFG_SETS Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Root cause:When execute CFG_SET_1, the value MBDSR1 and MBDSR0 was set 0x3. Then execute CFG_SET_2: the value MBDSR1 and MBDSR0 was not reset to 0x0 and in Can driver only implemented setting MBDSR0 and MBDSR1 bits, not clear these bits (line1049 ? 1105): .... case (FLEXCAN_16_BYTES_U32): #if (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_96_U8) if (FLEXCAN_MB_96_U8 == u8MessageBufferControllerSize[Controller]) { /* @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)0x1U<<FLEXCAN_MBDSR2_OFFSET); } #endif /* @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)1U<<FLEXCAN_MBDSR1_OFFSET); /* @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)1U<<FLEXCAN_MBDSR0_OFFSET); break; .... Therefore when execute CFG_SET_2, The value MBDSR1 and MBDSR0 still was 0x03(expected 0x1) Proposed solution: It need clear MBDSR0 and MBDSR1 bits before set them ex: ... case (FLEXCAN_16_BYTES_U32): #if (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_96_U8) if (FLEXCAN_MB_96_U8 == u8MessageBufferControllerSize[Controller]) { /*Clear FLEXCAN_FDCTRL*/ REG_BIT_CLEAR32(FLEXCAN_FDCTRL(u8HwOffset), (uint32)0x3U<<FLEXCAN_MBDSR2_OFFSET); /* @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),

Table continues on the next page...

ID	Subtype	Headline and Description
		(uint32)0x1U<<FLEXCAN_MBDSR2_OFFSET); } #endif /*Clear FLEXCAN_FDCTRL*/ REG_BIT_CLEAR32(FLEXCAN_FDCTRL(u8HwOffset), (uint32)0x3U<<FLEXCAN_MBDSR1_OFFSET); REG_BIT_CLEAR32(FLEXCAN_FDCTRL(u8HwOffset), (uint32)0x3U<<FLEXCAN_MBDSR0_OFFSET); /* @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)1U<<FLEXCAN_MBDSR1_OFFSET); /* @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)1U<<FLEXCAN_MBDSR0_OFFSET); ...",'Can_Flexcan.c
ENGR00371777	Defect	[CAN] The value of max_Mb is generated incorrectly for FlexCAN_B in Can_Cfg.h 'Problem detailed description (how to reproduce it): - Configure one project with: + CanHwChannel is FlexCAN_B (64MB) + CanRxProcessing and CanTxProcessing: INTERRUPT => The max_mb of FlexCAN_B is generated incorrectly: #define CAN_MB_CONTROLLER_2 96U - Execution test with scenario as below: Call Can_Init(); ... => The program will jump to IOVRx interrupt. Root cause: When max_mb of FlexCAN_B is 96. The register CAN_IMASK3 and CAN_IFLAG3 will be written when Can_Init is called. But the register CAN_IMASK3 and CAN_IFLAG3 do not exist in FlexCAN_B. Therefore The program will jump to IOVRx interrupt. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional Can_TC_1003 in Can_TS_53 Trigger: Execution Observed behavior: The program jump to IOVRx interrupt. Expected behavior: The program does not jump to IOVRx interrupt. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):",'Can_Cfg.h
ENGR00376016	NewWork	[CAN] Udate variable file for document CAN 'NewWork Description: In the UM has some label may be use generic between FlexCan and Mcan. But at the moment, they are defined as for MCAN (Example:

Table continues on the next page...

ID	Subtype	Headline and Description
		<IPV>_FD_PROPSEG_OFFSET) Expected behavior: UM is generated successfully and imformationin UM is correctly.",'update variable.xml
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): [...]",'Can_Flexcan.c
ENGR00379264	NewWork	[CAN] Update UM with description of timeout functionality and recommendations for configuring timeout parameters 'NewWork Description: Update UM with description and recommendations for configuring timeout parameters. Expected behavior: Update the description for timeout value field in the plugin to include a recommended value Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add a recommended value for the timeout value from the plugin",'Introduce to configure TimeOut for driver.
ENGR00363918	NewWork	[CAN] Update interrupt macro to cover more interrupts 'NewWork Description: Update interrupt macro in Can_Flexcan.h to cover all interrupts depend platform. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA",'Update interrupt macro in Can_Flexcan.h.
ENGR00364192	NewWork	[CAN] Update interrupt macro to cover more interrupts

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: Update interrupt macro in Can_Irq.c to cover all interrupts depend platform.','Update Can_Irq.c.
ENGR00373875	NewWork	[CAN] Update the generation file of ASR403 follow new data structure. 'NewWork Description: The data structure was changed. All ASR parameter have "PC to all" attribute which will be put to "CanStatic_..." structures, all the non-ASR parameter have "PC to PC\PB to PB" attribute which will be put to "Can_..." structure. This cause make the generation file (Can_Cfg.c anh Can_PbCfg.c) shall changes in order to compatible with new data structure. Expected behavior: the generation file of ASR403 shall compatible with new data structure. The generation and build shall be success. Requirement source: ASR403 Proposed solution (Optional): Update the generation file of the ASR403 follow the new data structure. Use the generation file of ASR421 in order to refer.","'Can_Cfg.c Can_PbCfg.c Can.c Can.h",ENGR00373926
ENGR00373392	NewWork	[CAN] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Conclusion: 1) Nodes that have constant false editable conditions (<a:a name="EDITABLE" value="false"/>) will change to (<a:a name="READONLY" value="true"/>) 2) Exception: the 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/> 3) Nodes that have EDITABLE attributes with type xPath will not be changed to

Table continues on the next page...

ID	Subtype	Headline and Description
		READONLY",'update Can.xdm
ENGR00367550	Defect	[CAN] Using improper solution to configure bits in register 'Problem detailed description (how to reproduce it): Please check again the following code: 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. 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): Using REG_RMW32 instead of a couple of REG_BIT_CLEAR32 and REG_BIT_SET32",'Can_Flexcan.c
ENGR00378363	NewWork	[CAN] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA",'Fix MISRA violations
ENGR00364241	NewWork	[CAN] 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,

Table continues on the next page...

ID	Subtype	Headline and Description
		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_SP EC_VERSION_PATCH"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJO R"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO R"!][!// [!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",'Add Can_VersionCheck.m. after that include this macro into template files
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

Table continues on the next page...

ID	Subtype	Headline and Description
		FdTimeSegment1 This check can however be removed completely since it is redundant with the ValueRange of CanControllerSeg2 in Can.xdm.","'flexcan/Can_Cfg.c flexcan/Can_PBcfg.c flexcan_42/Can_Cfg_42.c flexcan_42/Can_PBcfg_42.c mcan/Can_Cfg.c mcan/Can_PBcfg.c
ENGR00375862	Defect	[CAN]Fix LDRA error related the define user mode. 'Problem detailed description: Have one error when run LDRA. this error related the define for user mode. Observed behavior: All define for user mode in the Can., Can_FlexCan.c and Can_FlexCan.h Expected behavior: don't have error when run LDRA","'Can.c Can_FlexCan.c Can_FlexCan.h",ENGR00375866
ENGR00367057	Defect	[DIO] Fix MISRA errors 'Problem detailed description (how to reproduce it): In file Dio_Cfg.h, Misra violation comment for rule 19.15 (preventing the content of a header file from being included twice) for inclusion of file MemMap.h is having a typo: reference 'DIO_CFG_H_REF_2' is used instead of 'Dio_Cfg_H_REF_2'. In file Dio_Cfg.c, a cast to unsigned long ('UL') should be added in the macro 'reverse_bits' in order to prevent a Misra violation. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Misra violations occured Expected behavior: No Misra violations reported by PCLint Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):",'Update driver to fix misra error
ENGR00381461	Defect	[DIO] Fix MISRA violation which is reported by new version of MISRA tool 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate both rule 11.1 and 11.3, only rule 11.1 is reported. When issue in tool is fixed, we have to fix or comment the violations related to these rules. Because of new version of MISRA tool, MISRA rule 11.3 is reported on many lines of Dio.c and MISRA rule 5.1 violation in Dio.h

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: Casting is used between pointer and integer Test Case ID (internal TC that caught the defect) - optional NA Trigger: Use label tools as below: BLN_CODE-ANALYSIS_01.18.01 BLN_MISRAVIEW_01.09.01 Observed behavior: There are MISRA errors Expected behavior: There should be no MISRA errors unfixed or uncommented. Proposed solution (Optional): Add MISRA comments",'Add comments at the lines where make errors.
ENGR00378706	Defect	[DIO] Fix code review findings 'Problem detailed description (how to reproduce it): There are some findings after performing review code checklist for Matterhorn RTM 1.0.1 (ENGR00377414) Please see list of findings below: * SMCALRule 2.5 - Are the comments correct, complete and necessary?: --> In Dio_Cfg.h, the description for the define DIO_NUM_CHANNELS_PER_PORT_U16 is not exactly correct. * SMCALRule 2.27 - Last line in a file shall be an empty line. Is this rule fulfilled? --> In Dio_Cfg.h/.c. Dio_Ipw.h, Dio.mak, There is not an empty line at the end of these files Preconditions: NA Test Case ID (internal TC that caught the defect) - optional N/A Trigger: NA Observed behavior: NA Expected behavior: Fix all findings. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please refer to the Proposed Correction column in Detailed_Findings tab of the attached file.",'Fix all findings in the attached file.
ENGR00371381	Defect	[DIO] Fix code review findings against checklist 'Problem detailed description (how to reproduce it): There is one finding after code review against checklist from CR:ENGR00370864 1. Put the end brace of the CONST(Dio_ConfigType, DIO_CONST) [!"node:name(.)"!] structure definition in Dio_Cfg.c on the same column with the start brace. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Trigger: N/A Observed behavior: Some of the rules in the checklist for the code review are violated Expected behavior: All applicable rules in the code checklist should be fulfilled 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 Detailed description paragraph above",'Fix findings code review against checklist for RacerunnerIS RTM 2.0.0
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. - 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.",'Update Det error description in Usermanual
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):

Table continues on the next page...

ID	Subtype	Headline and Description
		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];",'Review and correct memory allocation
ENGR00365667	Defect	 [DIO] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): 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 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 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: Please see the Problem Description 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
		Update: - [DRV]_VersionCheck_Inc.m - [DRV]_VersionCheck_Src.m - [DRV]_VersionCheck_Src_PB.m",'Correct version checking in configuration template files
ENGR00366730	Defect	[DIO] Missing MemMap declarations in Dio.c and Dio_Cfg.h 'Problem detailed description (how to reproduce it): Missing MemMap declarations in Dio.c: extern CONST(Dio_ConfigType, DIO_CONST) Dio_ConfigPC; and Dio_Cfg.h: all global variables declarations. this causes a link error when sda=all compiler option is used. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: Link error when sda=all compiler option is used. Observed behavior: Link error appears when sda=all compiler option is used. Expected behavior: No link error when sda=all compiler option is 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 MemMap declarations in Dio.c and Dio_Cfg.h
ENGR00372701	Defect	[DIO] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: In Dio.mak, there is hard-coded copyright: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics Expected behavior: Plugin files shall not contain hard-coded copyright. Expected behavior: Hard-coded copyright does not exixst in plugin files. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Dio.mak should be updated: From: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics

Table continues on the next page...

ID	Subtype	Headline and Description
		To: (c) Copyright M4_SRC_YEAR_ID M4_SRC_COPYRIGHTED_TO","'Hard-coded copyright need to be replaced to M4_SRC_COPYRIGHTED_TObecause plugin files shall not contain hard-coded copyright. Files Modified: Dio.mak
ENGR00373393	NewWork	[DIO] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.","'Use "READONLY" instead of "EDITABLE" in xdm file
ENGR00378358	NewWork	[DIO] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","'- Verify all Misra errors and comments - Update the comment of Misra following Comment column in MISRADeviation.xls",ENGR00379284

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00364233	Defect	[DIO] 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_SP EC_VERSION_PATCH"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJO R"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO R"!][!// [!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",'Version checking is missing in configuration template files
ENGR00378941	Defect	[DIO][PORT] Fix code review findings 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		There are some findings after performing review code checklist for Matterhorn RTM 1.0.1 (ENGR00377414) Please see list of findings below: DIO * SMCALRule 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. Is this rule fulfilled? --> In Dio_Siul2.c, A write operation in Dio_Siul2_WriteChannel function need to be protected by an exclusive area because it interfere with a read-modify-write operation in Dio_Siul2_FlipChannel.. PORT * SMCALRule 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. Is this rule fulfilled? --> In Port_Siul2.c, the read-modify-write operations on SIUL2_MSCR_ADDR32 register by these functions have not yet been protected by exclusive area. * SMCALRule 5.8 - All 16 bits variables that are not initialized shall be placed into SEC_VAR_NO_INIT_16 (place the variable inside the <MSN>_START_SEC_VAR_NO_INIT_16> and <MSN>_STOP_SEC_VAR_NO_INIT_16>). Is this rule fulfilled? --> In Port_Siul2.c, the variable at this line used the wrong memory allocation Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Please see detailed description above Expected behavior: Fix all findings. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please refer to the Proposed Correction column in Detailed_Findings tab of the attached file.','Fix all findings in the attached file.
ENGR00370127	NewWork	[ETH] Activate Elektrobit Tresos Fibex import 'Customer request to have probability to import configuration from Fibex. # Enable/Disable comimporter information within generated plugin.xml file ## PLG_USE_COMIMPORTER = false PLG_USE_COMIMPORTER should be true','To make plugin be probability to import configuration from Fibex
ENGR00378327	NewWork	[ETH] Add state machine diagrams for drivers 'NewWork Description: Add state machine diagrams for drivers: - For module state

Table continues on the next page...

ID	Subtype	Headline and Description
		- For channels/groups where applicable The state diagrams shall be added in Module_HLD/Behavioral View For additional information related to state diagrams see the attachment Expected behavior: Add state diagrams in UML model Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add machine diagrams",'Have state diagram for driver state and controller state.
ENGR00372021	Defect	[ETH] Change order of check parameter for Eth_ControllerInit 'Problem detailed description (how to reproduce it): In the current implementation, other parameters depend on CtrlIdx. Depend on compiler, there might be issue when access array with wrong CtrlIdx. Therefore, the check for valid value of CtrlIdx should be checked first. Preconditions: Call Eth_ControllerInit with wrong CtrlIdx Test Case ID (internal TC that caught the defect) - optional (with specific compiler) Trigger: NA Observed behavior: Exception 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 CtrlIdx before other parameters.",'Change order of check parameters in Eth_ControllerInit
ENGR00367810	Defect	[ETH] Change pPhysAddrPtr from P2CONST to CONSTP2CONST 'Problem detailed description (how to reproduce it): In the current implementation, the pPhysAddrPtr was used in many different functions. All the usages are similar, which refer to value constant. Therefore, using P2CONST and CONSTP2CONST will lead to the same behavior. However, using different type of pointer lead to confusion for other when reading code. When can it be observed? (at configuration time, at runtime, at compile time?) Review Expected behavior: All similar usage of pPhysAddrPtr should have same type Proposed solution (Optional): Update type of input for local variable, using CONSTP2CONST for all similar use.","'Update type of input variable for pPhysAddrPtr. Modified in file: Eth_Enet.c
ENGR00378110	Defect	[ETH] Compiler warning with linaro compiler 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		There is the following issue with linaro compiler reported in gcc bugzilla: https://gcc.gnu.org/bugzilla/show_bug.cgi?id=53119 and https://gcc.gnu.org/bugzilla/show_bug.cgi?id=39589 This prevent initialize same value for all element of a struct. To avoid this compiler warning, we need to add extra brace "{" for the initialization. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Have compiler warning with Linaro compiler Expected behavior: Do not have compiler warnings with all compilers Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add extra brace "{" to prevent compiler warnings",'Fix compiler warnings
ENGR00363949	Defect	[ETH] Correct misra violation 'Problem detailed description (how to reproduce it): There are some misra error due to change size of base address and type definition for some platform. Preconditions: Misra report Test Case ID (internal TC that caught the defect) - optional NA Trigger: Check misra Observed behavior: Violation in Misra log and unable to generate misra report Expected behavior: Able to generate misra report Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add proper comment because this is specific for ARM64 platform.",'Correct misra comments
ENGR00381264	Defect	[ETH] Correct misra violation 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate both rule 11.1 and 11.3, only rule 11.1 is reported. When issue in tool is fixed, we have to fix or comment the violations related to these rules. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		NA Trigger: NA Observed behavior: Have error messages prevent generating Misra report Expected behavior: Can generate Misra report without any error 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 violation if possible or comment the violations with deviation reasons",'N/A
ENGR00365710	Defect	[ETH] EthCtrlPhyAddress shall not be an integer 'Problem detailed description (how to reproduce it): Physical address (EthCtrlPhyAddress) of the controller needs to be stored as an integer e.g. 0x665544332211. Normally a MAC address looks like 66:55:44:33:22:11. MAC address with an integer: * It is not specified in SWS. (see below SWS req.) * This deviation is not documented in IM or UM. * No verification of an valid MAC address is possible with an integer by tresos. * User unfriendly - It is not the common understanding of a MAC address how it has be look like According to ASR4.2 AUTOSAR_SWS_EthernetDriver.pdf (http://www.autosar.org/fileadmin/files/releases/4-2/software-architecture/communication-stack/standard/AUTOSAR_SWS_EthernetDriver.pdf) page 60 SWS item: \[ECUC_Eth_00020] : Name: EthCtrlPhyAddress Description: Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order. Regular Expression: [0-9a-fA-F]{2}[[:-][0-9a-fA-F]{2}]{5} Multiplicity: 0..1 Type: EcucStringParamDef Default: value -- maxLength: 17 minLength: 17 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): It can be done very easy in this way: <v:var name="EthCtrlPhyAddress" type="STRING"> <a:a name="DESC" value="EN: Specifies the unique 48-bit physical address (MAC address) of the controller in network byte order. String must match the regular expression:

Table continues on the next page...

ID	Subtype	Headline and Description
		[0-9a-fA-F]{2}[[:-][0-9a-fA-F]{2}]{5}"/> <a:da name="INVALID" type="XPath" expr="text:match(., '^[0-9a-fA-F]{2}([:-][0-9a-fA-F]{2}){5}$')" false="Physical address has an invalid value."/> </v:var>","'Update the type of EthCtrlPhyAddress in configuration. Modified in: Eth.xdm, Eth_GetRegister.m, Eth_Cfg.h, Eth_PBcfg.c, Eth_Lcfg.c
ENGR00363857	Defect	[ETH] Ethernet Configuration structure incorrect for VariantPostBuild configurations 'Problem detailed description (how to reproduce it): If the Ethernet module is configured in VariantPostBuild one or more EthConfigSet nodes can be configured. Here the quote from the Eth User manual: The EthConfigSet container contains one or more child nodes for each of the multiple configurations. ... However, only one constant (array) is generated in the source code: CONST(Eth_ConfigType, ETH_APPL_CONST) EthConfigSet[[!"num:i(count(EthConfigSet/*))"!]] = { ... This way it is not possible for the EcuM to set the correct reference for the Eth_Init() function. The EcuM only has a reference to the dedicated EthConfigSet. If there are e.g. 2 EthConfigSets named EthConfigSet_0 and EthConfigSet_1, and in EcuM the EthConfigSet_1 is referenced, it will call Eth_Init(&EthConfigSet_1); in EcuM_DefaultInitListOne. This will result in a compile error because there is no constant EthConfigSet_1 (only EthConfigSet[0] and EthConfigSet[1]). 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): NA",'Change in all generate template file,ENGR00373859
ENGR00371473	Defect	[ETH] Exception occured when calling Eth_ControllerInit without Eth_Init 'Problem detailed description (how to reproduce it): Error happens in Eth.c line 388. When Eth_ControllerInit() is called without Eth_Init() function previously, so Eth_InternalCfgPtr is NULL_PTR. Therefore, IOVR2 error will happen when using this pointer to point to memory region. Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Set Configvariant as PB or LT - Call Eth_ControllerInit() without Eth_Init() previously Observed behavior: IOVR2 error happens When can it be observed? (at configuration time, at runtime, at compile time?) Runtime Expected behavior: No error happen when call Eth_ControllerInit() without Eth_Init() previously Proposed solution (Optional): Check Ethernet state before checking this pointer",'Change order of checking input parameter of function Eth_ControllerInit. Modified in file: Eth.c
ENGR00374301	Defect	[ETH] Fix compiler warnings 'Problem detailed description (how to reproduce it): In file Eth.c, there was incorrect casting type for eReturnValue. Therefore, it leads to compiler warnings at this line. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler warnings Expected behavior: Compiler warnings does not appeared. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): cast to proper type.",'Fix compiler warnings
ENGR00374349	Defect	[ETH] Fix misra violation 'Problem detailed description (how to reproduce it): There are some misra violations in Eth driver. Please correct the violation. Preconditions: Generate misra report 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):",'Correct misra violation or comments un-prevent case
ENGR00378346	NewWork	[ETH] 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: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Analyze and update usage of exclusive area.,ENGR00379273 ENGR00381096

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00375534	NewWork	[ETH] Implementation of TxConfirmation and RxIndication should be generic based on IPVault 'NewWork Description: In current implementation, handling of TxConfirmation and RxIndication is specific for each platform. However, the implementation could be generic based on IP. Therefore, it is difficult to maintained for all platform with the same IPVault. It is better to move the implementation of these handling function to IP layer which is used as generic for similar IPVault. Expected behavior: RxConfirmation and RxIndication are in IPVault files Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Create new API in IPVault for handlings TxConfirmation and RxIndication and update the specific function to wrap only. Review again the functions which are not necessary to be global function then change them to static functions.",'Move function TxConfirmation and RxIndication to IPV files,ENGR00375536
ENGR00368171	NewWork	[ETH] 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.",'Add description about DEM/DET errors in UM
ENGR00362982	Defect	[ETH] Incorrect DET ID reported in Eth_ControllerInit 'Problem detailed description (how to reproduce it): In function Eth_ControllerInit, it check the CfgIdx after checking for CtrlIdx. So the pointer will point to an other region which might be NULL. So the check will report invalid CtrlIdx although we call with correct CtrlIdx. Preconditions: DET is ON Has only 1 config set for 1 controller. Call Eth_ControllerInit (0, 255) Test Case ID (internal TC that caught the defect) - optional ETH_TC_0027 Observed behavior: DET error ID is ETH_E_INV_CTRL_IDX but it should be ETH_E_INV_CONFIG Expected behavior: DET error ID is ETH_E_INV_CONFIG Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Check valid value for CfgIdx before check CtrlIdx.",'Correct the DET ID

Table continues on the next page...

ID	Subtype	Headline and Description
		reported. Modified in Eth.c
ENGR00362376	NewWork	[ETH] Issue while casting address to uint32 in 64-bit platform 'NewWork Description: In 64 bits platform, casting an address to uint32 cause issue. The compiler does not allow to do so. Therefore, please change the address from uint32 to uint64 for 64-bit platform. Expected behavior: Driver work with 64 bits microcontroller. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Using different line of code for 32 and 64 bits platfrom.",'Update config structure to store address as uint64 instead of uint32.
ENGR00372031	Defect	[ETH] Missing M4 tag to separate code for 4.0.3 and 4.2.1 'Problem detailed description (how to reproduce it): There are some code in ASR 4.2.1 was generated to plugin because of missing M4_Tag. These code sections do not impact to any functionalities. However, this cause issue when running LDRA reports. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: Build with CCOV option Observed behavior: Can' build for CCOV Expected behavior: Able to build with CCOV to generate LDRA reports. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add M4_Tag to remove code not used in ASR 4.0.3",'Remove code not use in ASR 4.0.3
ENGR00366773	Defect	[ETH] Missing MemMap declarations in driver files 'Problem detailed description (how to reproduce it): The following variables do not have consistent memory section guarding for definition and all other extern declarations: Eth_InternalCfgPtr Fls_ACEraseRomEnd Fr_FrDemCtrlTestResultPtr Lin_Config_PC Mcu_Config_PC Port_pConfig Wdg_au8Index Wdg_pConfigPC

Table continues on the next page...

ID	Subtype	Headline and Description
		the variables shoudl be allocted in the same MemMap.h section where it is defined, and in every other place where it's declared external. this causes a link error when sda=all compiler option is used. 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): Allocate proper memory sections for variables",'Add MemMap declarations in Dio.c and Dio_Cfg.h
ENGR00362528	Defect	[ETH] Possible issue while read or write address without proper alignment 'Problem detailed description (how to reproduce it): Eth_Enet_McastPoolAddItem is proper in PPC platform. However, while running test in ARM64 bit platform, there is issue with this function at read16 and write16 operation. The issue might because of address alignment. For example: /** @violates @ref Eth_Enet_c_REF_14 MISRA rule 11.4 */ /** @violates @ref Eth_Enet_c_REF_5 MISRA rule 17.4 */ /*REG_WRITE16(&Eth_ENET_rMulticastPool[u8CtrlIdx] [u16PoolIdx].au8PhysAddr[0], 0xAAAAU);*/ /* Write lower 4B of physical address */ /** @violates @ref Eth_Enet_c_REF_14 MISRA rule 11.4 */ /** @violates @ref Eth_Enet_c_REF_5 MISRA rule 17.4 */ /*REG_WRITE32(Eth_ENET_rMulticastPool[u8CtrlIdx] [u16PoolIdx].au8PhysAddr+2U, *((P2CONST(uint32, AUTOMATIC, ETH_APPL_DATA))(pPhysAddrPtr + 2U)));*/ /* Increment the count of used items */ Preconditions: Enable ETH_UPDATE_PHYS_ADDR_FILTER Test Case ID (internal TC that caught the defect) - optional ETH_TC_0101.c Trigger: NA Observed behavior: System Interrupt Expected behavior: Not have issue in any other platform. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Using byte access in stead of word acess.",'Prevent issue of allignment when access words in some platforms
ENGR00380324	NewWork	[ETH] Redundant checking with Eth_<IPV>_Transmit API

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: In Eth_<IPV>_Transmit API, it is check the status of remaining T01 bit in buffer descriptor. However, it is not necessary to check this because the condition for process this already implement in other places (in Eth_<IPV>_IsTxBufferLocked BT). Therefore, the redundant should be removed to optimize driver and improve performance. More detail about the use case about this improvement is described in Eth_Cases_Diagram.txt in Attachments tab. Requirement source:N/A Proposed solution (Optional): Remove these codes in Eth_Fec_Transmit(): SchM_Enter_Eth_ETH_EXCLUSIVE_AREA_00(); u32BDStatus = Eth_Fec_ReadTxBDBits(u32BDAddr, Eth_u8ActiveTxBD[u8CtrlIdx]); if(FEC_TXBD_TO1_U32 == (FEC_TXBD_TO1_U32 & u32BDStatus)) { ... } else { SchM_Exit_Eth_ETH_EXCLUSIVE_AREA_00(); }','Remove the redundant in the drivers.
ENGR00361974	Defect	[ETH] Resolve Error in VSMD report 'Problem detailed description (how to reproduce it): After implement muticontrollers, there was a VSMD error related to EthCtrlIdx. EcucSws_2027: MIN and MAX is mandatory for integer parameters. The node / TS_T2D35M10I1R0/Eth/EthConfigSet/EthCtrlConfig/EthCtrlIdx needs a MIN and MAX attribute. This is mandatory in VSMD. Actually, the min, max was checked properly in the attribute INVALID followed this requirement. However, the tools does not recognize the expression of the check. To prevent this error in the VSMD report, please separated the check for value follow ASR and do check for specific hardware in RANGE attribute. Refer to my proposal solution. This violation impacts to VSMD report only, it does not impact to functionality or application. Preconditions: VSMD report Test Case ID (internal TC that caught the defect) - optional NA Trigger: VSMD Report Observed behavior: Error with node EthCtrlIdx Expected behavior: There isn't error with EthCtrlIdx Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): <a:da name="RANGE" type="XPath"> <a:tst expr="text:uniq(node:fallback(../../*/EthCtrlIdx, text:split('1 2 3')), node:fallback(.,1))" false="Duplicated value. EthCtrlIdx must be unique for a configset"/>

Table continues on the next page...

ID	Subtype	Headline and Description
		<a:tst expr="num:i(.) < num:i(../../../../../EthGeneral/EthMaxCtrlsSupported)" false="EthCtrlIdx must be less than or equal to supported controllers number configured in EthMaxCtrlsSupported." /> </a:da> <a:da name="INVALID" type="Range"> <a:tst expr=">=0"/> <a:tst expr="<=255"/> </a:da>",'Update xdm to prevent VSMD error reported.
ENGR00373892	NewWork	[ETH] Split Configuration structure to Precompile and PostBuild variant 'NewWork Description: TWG decide to separate Configuration Structure to Precompile (for variant which is PreCompile for all implementation) and PostBuild (for variant which is PostBuild for implementation of PostBuild). Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Update config structure
ENGR00374366	Defect	[ETH] The address for RxBuffer data is over range when enable RxBuffer Wrap 'Problem detailed description (how to reproduce it): In Eth_Enet_ConfigureRxBuffers, calculate Rx buffer memory wrongly when enable Rx frame wrap and received buffer size equal to 1536, the current implementation as below: u32BufAddress += (VAR(uint32, AUTOMATIC))((((1536U / 2U) + ((VAR(uint32, AUTOMATIC))u16BufLength - 1U)) / u16BufLength) * u16BufLength); It does not need to allocate 1 more buffer when using Rx frame wrap because max frame length just 1536 bytes. Preconditions: - Enable EthUseMultiBufferRxFrames and EthEnableRxFrameWrap - Configure Rx buffer length is equal to 1536 Test Case ID (internal TC that caught the defect) - optional Eth_TC_1001 Observed behavior: Rx buffer memory is conflicted with variable memory Proposed solution (Optional): In case 1536 bytes, does not allocate 1 more buffer, just allocate 1 more buffer with buffer size smaller than 1536 bytes",'Update computation for leading memory use when multi-Rx buffer and Multi-Rx buffer wrap is enable
ENGR00371976	Defect	[ETH] The counter ENET_RMON_R_DROP does not exist on ENET controller 'Problem detailed description (how to reproduce it): Define ENET_RMON_R_DROP_ADDR16 is wrong because ENET_RMON_R_DROP does not exist on ENET controller. Therefore, it could cause IOVR1 when reading or writing into this memory location. Preconditions: N/A

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect): Eth_TC_0156 Trigger: N/A Observed behavior: IOVR1 when reading into this memory location Expected behavior: Can not access to this memory location Proposed solution (Optional): Remove define ENET_RMON_R_DROP_ADDR16 Use ENET_IEEE_R_DROP instead of ENET_RMON_R_DROP in function Eth_Enet_GetEtherStats() and Eth_Enet_MainFunction()",'Remove definition of ENET_RMON_R_DROP_ADDR16 in Eth_Enet_Counters.h and replace this by ENET_IEEE_R_DROP_ADDR16 in Eth_Enet.c
ENGR00373394	NewWork	[ETH] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Use cases: Conclusion: 1) Nodes that have constant false editable conditions (<a:a name="EDITABLE" value="false"/>) will change to (<a:a name="READONLY" value="true"/>) 2) Exception: the 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/> 3) Nodes that have EDITABLE attributes with type xPath will not be changed to READONLY",'EDITABLE node with constant condition will be changed to READONLY node.
ENGR00369040	Defect	[ETH] Wrong attribution for some nodes in xdm file 'Problem detailed description (how to reproduce it): When using a configuration which has EthCtrlPhyAddress value differs default value to generate project with EPD schema, the result is wrong because it did not generate EthCtrlPhyAddress value in Eth_PBcfg.c exactly. Developer's comments: Not only for node EthCtrlPhyAddress, for some other nodes under ASR 4.2.1, the implementation follow template file and does not

Table continues on the next page...

ID	Subtype	Headline and Description
		reflect the ASR model. Preconditions: Configure one EthCtrlPhyAddress which have EthCtrlPhyAddress equal: 66:55:44:33:22:11 Test Case ID (internal TC that caught the defect) - optional Eth_TS_epd_01 Trigger: Using Generate step. Observed behavior: Configured MAC Address is 00:00:00:00:00:00 static CONST(Eth_CtrlCfgType, ETH_APPL_CONST) EthConfigSet_0_EthCtrlConfig = { /* Configured MAC Address is 00:00:00:00:00:00 */ 0x00000000U, /**< First 32 bits of MAC address */ 0x00008808U, /**< Last 16 bits of MAC address<<16 | 0x8808 */ .... Expected behavior: Configured MAC Address must be 66:55:44:33:22:11 static CONST(Eth_CtrlCfgType, ETH_APPL_CONST) EthConfigSet_0_EthCtrlConfig = { /* Configured MAC Address is 66:55:44:33:22:11 */ 0x66554433U, /**< First 32 bits of MAC address */ 0x22118808U, /**< Last 16 bits of MAC address<<16 | 0x8808 */ .... Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Correct it in *.epd",'Change attribute for EthCtrlPhyAddress and DEM reference nodes (for ASR 4.2.1 nodes)
ENGR00366976	Defect	[ETH] Wrong path for check of parameter EthPhyInterface 'Problem detailed description (how to reproduce it): customer configured the mode RMII in the configuration parameter EthPhyInterface of the Eth module generated ENET_RCR register value is 0x40004004U. But according the reference module of the treerunner the bit RMII_MODE (bit 8) should be set. The root cause for this problem seems that the path of the check for parameter EthPhyInterface in the file Eth_GetRegisterValues.m is not correct: [!IF "EthVendorSpecific/EthPhyInterface = 'RMII'"!][!// but should be: [!IF "EthPhyInterface = 'RMII'"!][!// Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Can not configure for RMII mode. Expected behavior: Able to configure for RMII mode 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): Correct the mentioned line as: [!IF "EthPhyInterface = 'RMII'"!][!//",'Update Eth_GetRegister.m in order to have correct path
ENGR00365926	NewWork	[ETH] add support for Autosar standard 4.2.1 - Part 2 'NewWork Description: The original CR already implement some new functions, requirement. However, the configuration need to update base on discussion of TL and PM. We need to implement the change in xdm. And also for some missing update or other in order to compatible with update version. Note: the feature related to offloading and timestamp are not fully implemented because the target hardware does not support these feature. Expected behavior: Implement deltas to support the ASR 4.2.1 standard Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Review and implement changes in code to support Autosar 4.2.1 standard.",'Support ASR 4.2.1
ENGR00380955	Defect	[FEE] Data corruption occurs at swap if inconsistent blocks exist 'Problem detailed description (how to reproduce it): Scenario 1: - an INCONSISTENT block exists - swap occurs once - swap occurs second time, the INCONSISTENT block was not updated meanwhile - global cluster header pointer becomes corrupted - any subsequent write fails, by attempting to write the block header at wrong offsets in flash. - cluster swap occurs for any write attempt, if module is not reinitialized Scenario 2: - LEGACY_MODE==OFF - Immediate and INCONSISTENT block exists before swap - swap occurs - blocks data pointer become corrupted for blocks number above the INCONSISTENT one - data becomes corrupted for those blocks Steps to reproduce Scenario 1: - write all blocks once - change a block status to INCONSISTENT(by interrupting the write or simulating ECC errors) - force/make two swaps, by writing different blocks than the INCONSISTENT one - attempt to write any new block, FEE will try to write the new block header over the last previously written block header. - FEE will incorrectly update uHdrAddrIt at swap, thus making the underlying FLS driver to write an address smaller than the correct one and attempting to write flash already written

Table continues on the next page...

ID	Subtype	Headline and Description
		- write job will fail, DET can be called by FLS if configured, JobErrorNotification can be called if configured, a new cluster swap will be scheduled. Steps to reproduce Scenario 2: - LEGACY_MODE == OFF - a block is immediate and INCONSISTENT - a swap occurs - read a block number above the INCONSISTENT one, data should be corrupted Preconditions: Scenario 1: - an INCONSISTENT block exists - swap occurs once - swap occurs second time, the INCONSISTENT block was not updated meanwhile Scenario 2: - LEGACY_MODE == OFF - a block is immediate and INCONSISTENT - a swap occurs - a block with a number above the INCONSISTENT one after swap, is read before being written Test Case ID (internal TC that caught the defect) - optional NA Trigger: Customer Observed behavior: Scenario 1: Write job will fail, DET can be called by FLS if enabled and if FlsWriteBlankCheck is enabled, otherwise an ECC error is generated by the flash overprogram. JobErrorNotification will be called and a new cluster swap will be scheduled for the next write attempt. Scenario 2: Read data is corrupted for certain blocks after swap. Expected behavior: No read or write error should occur, Fee_JobErrorNotification should not occur, no DET should be reported by the FLS driver. 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 update of uHdrAddrIt and uDataAddr at swap, in function Fee_JobIntSwapClrVldDone(). The update of both header and data pointer at the end of the cluster swap, should be made for all the cases when data is allocated during swap.",'Changed Fee.c to correct the update of uHdrAddrIt in Fee_JobIntSwapClrVldDone also for the case the block is inconsistent and Fee_aBlockInfo[uBlockIt].uDataAddr is 0.
ENGR00378342	Defect	[FEE] Implement data consistency mechanism (exclusive areas) 'Problem detailed description (how to reproduce it): Missing exclusive areas to ensure thread-safety for high level API functions, needed to protect the internal variables in: Fee_Read, Fee_Write, Fee_InvalidateBlock, Fee_EraseImmediateBlock. Exclusive areas are needed to ensure thread-safety for the high level api functions.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: The above listed API functions interrupt each other right after checking and before setting the status variable. 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 exclusive areas for the high level API functions.",'Add exclusive areas for high level API functions.
ENGR00368166	NewWork	[FEE] 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.",'Improved Dem/Det error description in Usermanual
ENGR00368578	Defect	[FEE] 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 "Fee_aReservedAreaTouched" variable is placed in "FEE_START_SEC_VAR_NO_INIT_UNSPECIFIED" memory section instead of "FEE_START_SEC_VAR_INIT_32" section. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: "Fee_aReservedAreaTouched" variable is placed in "FEE_START_SEC_VAR_NO_INIT_UNSPECIFIED" memory section instead of "FEE_START_SEC_VAR_INIT_32" section. 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.) Proposed solution (Optional): Move "Fee_aReservedAreaTouched" variable from "FEE_START_SEC_VAR_NO_INIT_UNSPECIFIED" to "FEE_START_SEC_VAR_INIT_32" section.","'Move "Fee_aReservedAreaTouched" variable from "FEE_START_SEC_VAR_NO_INIT_UNSPECIFIED" to "FEE_START_SEC_VAR_INIT_32" section.
ENGR00379746	NewWork	[FEE] Keep not defined FEE blocks at cluster swap 'NewWork Description: Add support for swaping foreign blocks at cluster swap. The "FEE Swap foreign blocks" feature was developed to support the following customer use case: the customer has two different projects, BOOTLOADER and APPLICATION, which are separately compiled, built into different executable files which are run on the same ECU in different scenarios. This means the two executables share the same data flash emulated for calibration data. For the two projects the customer uses two different FEE block configurations: for the BOOTLOADER project some blocks, for APPLICATION project other blocks, maybe some blocks are common. In the previous FEE implementation it was not possible to run over same data flash with different FEE configurations because FEE used to keep at cluster swap only blocks which were present in the current configuration. This behavior had the disadvantage that each time a new block was added in the APPLICATION project, the customer was required to update also the FEE configuration from the BOOTLOADER project, even if BOOTLOADER blocks were not added or deleted. Requirement source: Customer request Proposed solution (Optional): Add support for swaping foreign blocks at cluster swap.",'Implement new feature to allow swapping of foreign blocks at cluster swap.
ENGR00373216	Defect	[FEE] Notifications don't check that Fee is initialized 'Problem detailed description (how to reproduce it): In AUTOSAR architecture all memory stack init (Nvm,MemIf,Fee,Fls)must be called at the same time in the system(startup 2)- see [SWS_EcuM_02563], so this defect should not be apparent to the user. If Fls_Init and Fls job request is called before the Fee_Init and the FLS job processing finishes before Fee_Init, then the FLS calls Fee job end notification or Fee job error notification before Fee is initialized and these 2 functions incorrectly update some Fee internal variables.

Table continues on the next page...

ID	Subtype	Headline and Description
		Propose fix: In the 2 Fee notifications add check that Fee is initialized: -If Fee UNINIT, then report to DET and do nothing else -If Fee not UNINIT, then do the normal processing 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): [...]","'In the 2 Fee notifications add check that Fee is initialized: -If Fee UNINIT, then report to DET and do nothing else -If Fee not UNINIT, then do the normal processing
ENGR00376199	NewWork	[FEE] Operations abort/resume mechanism is not supported in FEE driver 'According to CR ENGR00161338, Operations abort/resume mechanism is not supported in FEE driver. So, these requirements bellow should be rejected for FEE module : FEE180 If the current module status is MEMIF_BUSY_INTERNAL and if the internal management operation can be aborted without jeopardizing the data consistency: the function Fee_MainFunction shall save all information which is necessary to restart the internal management operation, abort the internal management operation and start processing the job requested by the upper layer. FEE181 If an internal management operation has been suspended because of a job request from the upper layer, the function Fee_MainFunction shall resume this internal management operation once the job requested by the upper layer has been finished. FEE182 If an internal management operation has been aborted because of a job request from the upper layer, the function Fee_MainFunction shall restart this internal management operation once the job requested by the upper layer has been finished.","'- Mask requirements FEE180, FEE181, FEE182 as rejected in DOORs - Update UM/IM
ENGR00364090	Defect	[FEE] Plugin generation is not aligned with MCAL standard 'Problem detailed description (how to reproduce it): 1. The file Fee.mak contains the generation of the Fee_Cfg.h, Fee_Cfg.c and Fee_PBcfg.c into the plugin directory "generate". The correct one must be "generate_PC". Please update the file Fee.mak to reflect the approach of the MCAL standard 2. The file Fee.mak has a tag "PLG_CUSTOM_GENS" that is empty. It must

Table continues on the next page...

ID	Subtype	Headline and Description
		contain the dirs/variants for which the generation of the plugin exists. For FEE, the correct tag must be "PLG_CUSTOM_GENS = generate_PC" The external impact resumes to a different directory name/path for the configuration files inside the FEE plugin. No other functional impact. 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): The files: Fee_Cfg.h, Fee_Cfg.c, Fee_PBcfg.c and Fee_VersionCheck.m should be generated to: "outdir=generate_PC" and "outdir=generate_PB", instead of: "outdir=generate".","'Updated "generate" folder name to "generate_PC" to reflect the approach of the MCAL standard
ENGR00374606	Defect	[FEE] Reference node missing to check existing or not 'Problem detailed description (how to reproduce it): Tresos Studio throws following errors and do not generate the files due to "node:ref(./FeeSectorRef)": Failed to run generator "Fee_TS_T2D47M9I0R0_GeneratorId" (mode: "generate") for module "Fee_TS_T2D47M9I0R0 The XPath-expression "$FeeClusterSize + num:i(node:ref(./FeeSectorRef)/FlsSectorSize) * num:i(node:ref(./FeeSectorRef)/FlsNumberOfSectors)" caused an error: (1844) No node found for path "" Parsing file "C:\work\Projects\63_ACP-7.5_S32R274_Conti\tresos\plugins\Fee_TS_T2D47M9I0R0/generate_PC/src/Fee_Cfg.c", line "124" The XPath-expression "$FeeClusterSize + num:i(node:ref(./FeeSectorRef)/FlsSectorSize) * num:i(node:ref(./FeeSectorRef)/FlsNumberOfSectors)" caused an error: (1844) No node found for path "" Failed to generate file "C:\work\Projects\63_ACP-7.5_S32R274_Conti\tresos\plugins\Fee_TS_T2D47M9I0R0/generate_PC/src/Fee_Cfg.c" The XPath-expression "$FeeClusterSize + (num:i(node:ref(./FeeSectorRef)/FlsSectorSize) * num:i(node:ref(./FeeSectorRef)/FlsNumberOfSectors))" caused an error: (1844) No node found for path "" Parsing file "C:\work\Projects\63_ACP-7.5_S32R274_Conti\tresos\plugins\Fee_TS_T2D47M9I0R0/generate_PC/include/Fee_Cfg.h", line "240" The XPath-expression "$FeeClusterSize + (num:i(node:ref(./FeeSectorRef)/FlsSectorSize) * num:i(node:ref(./FeeSectorRef)/FlsNumberOfSectors))" caused an error: (1844) No node found for path "" Failed to generate file "C:\work\Projects\63_ACP-7.5_S32R274_Conti\tresos\plugins\Fee_TS_T2D47M9I0R0/generate_PC/include/Fee_Cfg.h" Please can you analyze and fix it? Let to see the following code: [!LOOP "FeeCluster/*"!] [!VAR "FeeClusterSize" = "0"!] [!LOOP "FeeSector/*"!]

Table continues on the next page...

ID	Subtype	Headline and Description
		[!VAR "FeeClusterSize" = "$FeeClusterSize + (num:i(node:ref(./FeeSectorRef)/FlsSectorSize) * num:i(node:ref(./FeeSectorRef)/FlsNumberOfSectors))"!] [!ENDLOOP!] [!IF "$FeeClusterSize < $FeeSmallestClusterSize"!] [!VAR "FeeSmallestClusterSize" = "$FeeClusterSize"!] [!ENDIF!] [!ENDLOOP!] Fee can not refer to Fls sector in case user available to disable/enable "FeeSectorRef" 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): [...]",'Fix generation error when there is one/some disabled FeeSectorRef.
ENGR00378565	Defect	[FEE] Sector list reorderd alphabetically when generating the .epc with Tresos 'Initial Description: The sector list is reordered alphabetically when generating the .epc with Tresos. The issue does not appear when generating using the .xdm file. Because of this, the index used to calculate the validity of FlsSector will be wrong. This issue becomes problematic only when using more than 10 configured flash sectors. Problem detailed description (how to reproduce it): Configure more than 10 FeeSectorRef or change the default FeeSectorRef names. Since FeeSectorRef is using @index paramenter to calculate the checking condition, then FeeSectorRef will get error when reordering alphabetically occurs in the FeeSectorRef list. Preconditions: Configure FlsSector list with more than 10 sector or Customize FlsSector name, something like below : -FeeSecorRef_0 -FeeSectorRef_1 -FeeSectorRef_02 .... Observed behavior: - Generation failure because of wrong checking condition in FeeSectorRef node. The sector list is reordered in .epc files, and instead of: FlsSector_0 FlsSector_1 FlsSector_2 .... we get:

Table continues on the next page...

ID	Subtype	Headline and Description
		FlsSector_0 FlsSector_1 FlsSector_10 FlsSector_11 ... When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: Generation pass with more than 10 Fls sector in FeeSectorRef list Reported release baseline: Proposed solution (Optional): - Add a new Vendor specific parameter: "FeeSectorRefIndex" to use instead of "@index" to keep FeeSectorRef in logical order when checking various conditions. NewWork Classification: (internal task, improvement, feature request) NA In case for feature request, does a TWG ticket exist (cloned or linked)? TBC (Yes/No) NewWork Description: NA Expected behavior: Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)","'Add a vender specific FeeSectorIndex to FeeSector conainter to keep FeeSectorRef in the order. Also, Use FeeSectorIndex to calculate checking condition for FeeSectorRef instead of @index parameter
ENGR00376024	Defect	[FEE] UM deviation table and requirements status is not synchronized 'Problem detailed description (how to reproduce it): SWS_Fee_00180, SWS_Fee_00181, SWS_Fee_00182, according to CR : ENGR00161338, Fee driver doesn't implement them and in the UM, those requirements marked as Not implemented. But in DOORS, those requirements were masked as fulfilled. Preconditions: Analysis requirement for Fee module Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Deviation requirement table is not synchrous with requirement analysis on DOORS server Expected behavior: -the requirements which are marked as not implemented in the deviation table in UM are not fulfilled in DOORS and -the requirements which are Rejected/not fulfilled in DOORs appear in the UM deviation table Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): - Verify UM vs DOORS - Update design, traceability, test traceability","'- Remove FEE180, FEE181,

Table continues on the next page...

ID	Subtype	Headline and Description
		FEE182 requirements from Fee_MainFunction in UML design - Update UM : add these requirements as Not implemented : FEE130, FEE168, SWS_Fee_00187,ECUC_Fee_00153 - Masked FEE180, FEE181, FEE182 as not fulfilled in DOORS
ENGR00374172	NewWork	[FEE] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.","'Use "READONLY" instead of "EDITABLE" in xdm file for parameters which have Editable=FALSE in Fee.xdm : Changed parameters : - FeeNumberOfWriteCycles - VendorApiInfix - FeePollingMode
ENGR00364750	Defect	[FEE] 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"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJO

Table continues on the next page...

ID	Subtype	Headline and Description
		R"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO R"!][!// [!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",'-Add version check to FEE configuration templates files
ENGR00376293	NewWork	[FEE] When a cluster format is required during initialization a notification callback should be called 'NewWork Description: Fee shall call the configurable notification FeeClusterFormatNotification to inform the user in case a cluster format is triggered during the Fee initialization Customer use case: observed a scenario in which during the initialization of the Fee/Fls modules, if header information for the EEPROM blocks is erased or corrupted, that the module attempts to perform a cluster swap in order to properly initialize the memory blocks. the problem, because they have no way of knowing when this is occurring or when this will occur. This means that during each boot, every time initialize the Fee/Fls drivers, they need to synchronously execute the respective MainFunctions in order to process the cluster swap if it is occurring, and in many cases this can take from 1sec to 4secs. Ideally, when the a cluster swap is required during initialization, there is a callback, notification, or API that - they can call to get the status, and can then trigger this process to happen in the background. Expected behavior: Fee shall call the configurable notification FeeClusterFormatNotification to inform the user in case a cluster format is triggered during the Fee initialization Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		Customer Request Proposed solution (Optional): Fee shall call the configurable notification FeeClusterFormatNotification to inform the user in case a cluster format is triggered during the Fee initialization","When a cluster format is required during initialization a notification callback is called. Files changed: Fee.xdm Fee_cfg.h fFe.c
ENGR00373285	NewWork	[FLS] Add IM note about IVOR1 bahavior for DCache enabled 'NewWork Description: The IVOR1 behavior is different when DCache is on, regarding the handler return address. DCache off => syndrome = 0x00088008 => MCSRR0 will hold the address of the offending instruction, which has caused the exception. DCache on => syndrome = 0x00080008 => MCSRR0 will hold the address of the next instruction, which hasn`t got to execute yes. see attached emails for more details. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add IM note, in the lines of: "The flash driver requires data cache to be disabled. If despite that, data cache is enabled, although the flash area is defined as cache inhibited in SMPU, the bevavior of the the Machine Check(IVOR1) exception might be different. An asynchronous ECC exception might be triggered by a cache line fill prefetch, which would report a different syndrome and return address. The interrupt handler implementation should be updated accordingly." Create test(can be a duplicated ECC test case) in which to test with the following setup: Data cache enabled, flash area configured as Cache-Inhibited in the SMPU, and Mpc_getNextInstructionAddress() modified to accommodate the new return value: MCSRR0 for syndrome 0x00080008, instead of MCSRR0 +2/+4 for syndrome 0x00088008. Create test(can be a duplicated ECC test case) in which to test with the following setup: Data cache enabled, flash area configured as Cache-Inhibited in the SMPU, and MSR[ME] bit cleared. This would mask any Asynchronous interrupts and allow the execution of MachineChecks of type Error report.','Updated IM for data cache enabled use case.
ENGR00372839	NewWork	[FLS] Add redundant error monitoring for flash reads 'NewWork Description: Add redundant error monitoring for flash reads by checking C55FMC_MCR[RVE] and C55FMC_MCR[RRE] bits. These bits offer an extra check for failed reads, also already covered by ECC mismatch error. These bits might be useful in case of hardware erratas affecting the FCCU and AEE mechanism. Proposed solution: Check, after all flash reads(read, compare, verify erase, verify write), the

Table continues on the next page...

ID	Subtype	Headline and Description
		C55FMC_MCR[RVE] and C55FMC_MCR[RRE] bits. Clear all bits before a read. Put the current flash job as failed if any bit is set. Add an NonAutosar conditional compile check for this functionality(for ex: FlsEnableReadVoltageChecks).','Add extended read error check for RVE and RRE bits.,ENGR00373473
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 Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 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 Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 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

Table continues on the next page...

ID	Subtype	Headline and Description
		void Fls_Flash_AccessCode(P2VAR(volatile uint8, AUTOMATIC, FLS_CODE) regBasePtr, P2FUNC(void, FLS_CODE, CallBack)(void)); #endif Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 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","'Port flash access function declaration to GCC format, for linker section placement.",ENGR00369261
ENGR00362978	NewWork	[FLS] Add the missing M4_SRC_AR_RELEASE_REVISION generate file 'NewWork Description: The XPATH "../../FlsGeneral/FlsAcLoadOnJobStart" is wrong in ASR4.2.1. When you set the checkbox FlsAcLoadOnJobStart in TRESOS, the code that was generated will be wrong. So when you build the code, the error "Fls_Flash_AccessCode isn't declared" will occurs Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add the macro ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl for the following code in Fls_Cfg.c and Fls_PbCfg.c [!WS "4"!](Fls_Flash_AcErasePtrType)[!IF "../../FlsGeneral/FlsAcLoadOnJobStart"!][!IF "normalize-space(./FlsAcErasePointer) = 'NULL_PTR' or normalize-space(./FlsAcErasePointer) = 'NULL'"!][!"normalize-space(./FlsAcErase)"!]U[!ELSE!]&[!"./FlsAcErasePointer"!][!ENDIF!][!ELSE!]&Fls_Flash_AccessCode[!ENDIF!], /* FlsAcErase */ TO ifelse(M4_SRC_AR_RELEASE_REVISION,`ASR_REL_4_2_REV_0001',`dnl [!WS "4"!](Fls_Flash_AcErasePtrType)[!IF "../FlsGeneral/FlsAcLoadOnJobStart"!][!IF "normalize-space(./FlsAcErasePointer) = 'NULL_PTR' or normalize-space(./FlsAcErasePointer) = 'NULL'"!][!"normalize-space(./FlsAcErase)"!]U[!ELSE!]&[!"./FlsAcErasePointer"!][!ENDIF!][!ELSE!]&Fls_Flash_AccessCode[!ENDIF!], /* FlsAcErase */ ',`dnl [!WS "4"!](Fls_Flash_AcErasePtrType)[!IF "../../FlsGeneral/FlsAcLoadOnJobStart"!][!IF "normalize-space(./FlsAcErasePointer) = 'NULL_PTR' or normalize-space(./FlsAcErasePointer) = 'NULL'"!][!"normalize-space(./FlsAcErase)"!]U[!ELSE!]&[!"./FlsAcErasePointer"!][!ENDIF!][!ELSE!]&Fls_Flash_AccessCode[!ENDIF!], /* FlsAcErase */

Table continues on the next page...

ID	Subtype	Headline and Description
		')dnl","'Files modified: + Fls_Cfg.c + Fls_PBCfg.c
ENGR00374491	NewWork	[FLS] Change Cache macros name 'NewWork Description: The Cache macros("cache_invalidate_all", etc) had been moved from BASE to MCL, so the names had been changed. The cache functions are executed in supervisor mode if configuration parameter "Mcl Enable User Mode Support" is checked in MCL configuration. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. Update in driver the name for all calls of cache invalidation/flush defines: cache_flush_all -> Mcl_CacheFlushAll cache_invalidate_all -> Mcl_CacheInvalidateAll cache_flush_ps -> Mcl_CacheFlushPs cache_invalidate_ps -> Mcl_CacheInvalidatePs cache_flush_pc -> Mcl_CacheFlushPc cache_invalidate_pc -> Mcl_CacheInvalidatePc 2. Include MCL as a driver dependency, and enable "Mcl Enable User Mode Support" in its configuration, in order to execute the cache functions in supervisor mode. 3. Include "Mcl.h" in the file which uses the cache functions, for cache function prototypes and CACHE_INVALIDATE_MACROS define.",'Chache macros names changed and Mcl added as Fls dependecy.,ENGR00374493
ENGR00372796	NewWork	[FLS] Change Fls_Flash_SectorCompare implementation for ASR 4.2 usecase 'NewWork Description: Fls_Flash_SectorCompare function references a local variable in ASR 4.2 usecase, for BlankCheck Api. Change the local variable to a global variable in order to avoid any possible problems when referencing a local variable on stack. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Replace local variable "blankCheckValue" from Fls_Flash_SectorCompare function with global variable "Fls_Flash_ErasedValue"","'Replace local variable "blankCheckValue" from Fls_Flash_SectorCompare function with global variable "Fls_Flash_ErasedValue" defined in Fls_LLD_Const.m4
ENGR00381367	Defect	[FLS] Correct misra violation 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate both rule 11.1 and 11.3, only rule 11.1 is reported. When issue in tool is fixed, we have to fix or comment the violations related to

Table continues on the next page...

ID	Subtype	Headline and Description
		these rules. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Have error messages prevent generating Misra report Expected behavior: Can generate Misra report without any error Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): All of the rules are correct except several rule in MISRA deviation. If it cannot fix, comment violate rule above these line.','N/A,ENGR00381372
ENGR00373343	Defect	[FLS] DEM reference name must be the same as DEM error name 'Problem detailed description (how to reproduce it): The DEM reference name configured in FLS plugin, has to be the same as the Autosar DEM error name. For example: For Fls "FLS_E_COMPARE_FAILED" DEM parameter,inside "Fls Dem Event Parameter References", the DEM reference has to be: "/Dem/Dem/DemConfigSet_0/FLS_E_COMPARE_FAILED" Preconditions: The corresponding DEM parameter has to be enabled. Proposed solution (Optional): Fix codetemplate generator to use the configured DEM reference instead of the Fls DEM error node. Workaround: Configure for each Fls DEM error a reference which has the same name as the Autosar error name, the Fls DEM node error name. Test Case ID (internal TC that caught the defect) - optional NA Trigger: Observed behavior: Build failed when The DEM reference name configured in FLS plugin is not the same as the Autosar error name Expected behavior: User can configure DEM reference with the customized name Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Fix codetemplate generator to use the configured DEM reference instead of the Fls DEM error node. Workaround: Configure for each Fls DEM error a reference which has the same name as the Autosar","'The DEM reference name configured in FLS plugin, has to be the same as the Autosar DEM error name.
ENGR00368351	Defect	[FLS] FLS-Driver causes code execution from RAM 'Problem detailed description (how to reproduce it): The FLS-Driver contains one function.

Table continues on the next page...

ID	Subtype	Headline and Description
		* 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. Preconditions: FlsAcLoanOnJobStart parameters is enabled in the flash configuration. 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): Request1: * 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. Answer: Memory shall be allocated for every section defined in MemMap.h. The "Fls_Flash_AccessCode" function is placed/mapped in the linker section "acfls_code_rom". This is done automatically by the FLS driver and its BASE dependencies, it does not need any external operations. When executing exclusively from ROM, this section has to be placed/linked in a flash sector which resides in a different partition than the partitions which contain the sectors configured for erase/write operations. Also, when executing exclusively from ROM, the configuration option "Fls Load Access Code On Job Start" should be disabled. The call to "Fls_Flash_AccessCode" function is done by the driver internally, and there is no external action needed. The IM manual will be updated to include these points. Add a diagram to illustrate the two use cases of Fls_LoadAc function. See attached example. Request2: * In addition please take actions that this function cannot erase the flash block in which itself resides. Answer: It is an external responsibility to ensure that code sections or other sensitive sections, are not placed in the same sectors configured for use in the Flash driver configuration.",'Updated IM manual.
ENGR00375402	Defect	[FLS] File version check missing in some source files

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): "SOURCE FILE VERSION INFORMATION" and "FILE VERSION CHECKS" is missing in some source files. 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): Check all source files and add file version information and cross check, similar with: (example): /* Check if current file and Reg_eSys.h header file are of the same vendor */ #if (ESYS_FLASHC_VENDOR_ID != REG_ESYS_VENDOR_ID) #error "Reg_eSys_FLASHC.h and Reg_eSys.h have different vendor ids" #endif #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK /* Check if current file and Reg_eSys.h header file are of the same Autosar version */ #if ((ESYS_FLASHC_AR_RELEASE_MAJOR_VERSION != REG_ESYS_AR_RELEASE_MAJOR_VERSION) || \ (ESYS_FLASHC_AR_RELEASE_MINOR_VERSION != REG_ESYS_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Reg_eSys_FLASHC.h and Reg_eSys.h are different" #endif #endif /* Check if current file and Reg_eSys.h header file are of the same software version */ #if ((ESYS_FLASHC_SW_MAJOR_VERSION != REG_ESYS_SW_MAJOR_VERSION) || \ (ESYS_FLASHC_SW_MINOR_VERSION != REG_ESYS_SW_MINOR_VERSION) || \ (ESYS_FLASHC_SW_PATCH_VERSION != REG_ESYS_SW_PATCH_VERSION)) #error "Software Version Numbers of Reg_eSys_FLASHC.h and Reg_eSys.h are different" #endif",'Update File Version Checks.
ENGR00370945	NewWork	[FLS] Fls_DsiHandler doesn't support all ECC errors 'Problem detailed description (how to reproduce it): Fls_DsiHandler doesn't support all ECC errors which I can trigger on our data (by using winIDEA memory ECC corruption functionality or by toggling power supply during NvM operations). Here are the details from Fls_DsiHandler: ECC properties : instruction_ptr = Ptr(0x0120D01C) data_ptr = Ptr(0x00FA0008)

Table continues on the next page...

ID	Subtype	Headline and Description
		syndrome_u32 = 0x00080008 FLS properties: Fls_Flash_ReadStatus = FLASH_E_PENDING Fls_Flash_ReadAddressPtr = Ptr(0x00FA0000) (NvM block 1 start address) Function Fls_DsiHandler support only ECC errors with syndrome == 0x00088008 (I have syndrome equal 0x00080008 where LD bit is not set). Function Fls_DsiHandler support only ECC errors when data_ptr == Fls_Flash_ReadAddressPtr (I have exception from 0x00FA0008 but ReadAddressPtr is equal 0x00FA0000). Flash sector not repaired - function return with FLS_UNHANDLED ? Is this ECC error syndrome is unhandled on purpose? This scenario can disable the device permanently so this is most important for me. Preconditions: Data cache enabled by L1CSR0[DCE]=1, and memory flash region used by FLS driver is cache inhibited by setting bit CI=1 in SMPU (i.e. data cache restriction as described in integration manual is followed but it does not work). Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]","'<Description of changes, e.g. Check was added to ... The function xyz was modified because ...> Files Modified: <e.g. Adc_Irq.c > <...>",ENGR00372812
ENGR00378341	Defect	[FLS] Implement data consistency mechanism (exclusive areas) 'Problem detailed description (how to reproduce it): Exclusive areas used in the high level APIs have the same name. The exclusive area name should be unique per function. Exclusive areas used in the low level IP functions cannot cover all the intended use case scenarios(for example, they are not intended to protect a sequence from being interrupted). If interrupt mode is supported on specific platform which support this mode, the flash processing interrupt may trigger inside an exclusive area inside Fls_Write or Fls_Erase, if configured. Preconditions: For high level API exclusive areas: only some high level APIs are used or can interrupt each other in the applications use-case, and the same exclusive area has to be implemented and taken also for the APIs where it is not needed. For low level IP exclusive areas which guard the flash access: the sequence inside the exclusive area has to be protected to not be interrupted by any other code executed from the same read-while-write partition(for the program operations) or to not be interrupted by other master read accesses, when data flash sectors are used and ECC error reporting has to ensure exclusivity to the flash driver for the error bits. If interrupt mode is supported on the specific platform, the flash processing interrupt may trigger inside an exclusive area inside Fls_Write or Fls_Erase, if configured. Test Case ID (internal TC that caught the defect) - optional NA

Table continues on the next page...

ID	Subtype	Headline and Description
		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): 1. Rename exclusive areas from high-level APIs. Assign an unique name for each exclusive area per function. This way, it can be decided to protect only some specific APIs against thread concurrent access, thus it is only needed to implement a part of the exclusive areas by the higher layers. 2. Remove the exclusive areas used in the low level IP functions and replace them with notifications. That exclusive area was used to guard the high-voltage programming and the read access to flash memory. 4. Attach the expected output documents(template, IM update).",'Rename exclusive areas from high level API functions. Replace low level exclusive areas with notifications.,ENGR00379307
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.",'Added DET description in User manual.
ENGR00373414	Defect	[FLS] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in our MCAL drivers. Variable declarations should have the definitions and all extern declarations mapped to the same memmap sections. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: External declaration the configuration structure is not placed in the "FLS_START_SEC_CONFIG_DATA_UNSPECIFIED" memory section, as is the definition, thus the global variable is mapped to the default ".data" section. Expected behavior: All global variables used by Mcal drivers must be mapped to the special ".mcal_data" 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 below construct around the external declaration of the configuration structure: #define FLS_START_SEC_CONFIG_DATA_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		#include "Fls_MemMap.h"","'Add below construct around the external declaration of the configuration structure: #define FLS_START_SEC_CONFIG_DATA_UNSPECIFIED #include "Fls_MemMap.h" Only impact to Fls driver on ASR 4.2.1 driver
ENGR00373169	NewWork	[FLS] Investigate and implement to fix compiler warnings on Diab compiler 'NewWork Description: 1.Investigate the following compiler warning: (dcc:1824): explicit cast from 'ptr-to-volatile uint32' to 'volatile uint32' discards volatile qualifier, and similar. on Diab compiler, version 5.9.4.8 The new warning was added on version 5.9.4.7: ... Introduced new compiler warnings/errors generated by dcc: Warning 1824 ? Explicit cast discards volatile qualifier; 2.Investigate why some of these warnings are duplicated and reported again at the last line of the file. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Analyze any functional impact of the warnings and fix or comment if needed.",'Add compiler warnings comment for DIAB 5.9.4.8 compiler version.,ENGR00380207
ENGR00373902	NewWork	[FLS] Investigate and implement to reduce cyclomatic complexity and nesting level 'Requirements: - 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. 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 ---------- Implementation details: In AUTOSAR_MCAL_<MDL>_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments - for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column

Table continues on the next page...

ID	Subtype	Headline and Description
		- for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth >4 the problems shall be resolved",'- Remove interleaved implementation from IPV_FLASHV2,ENGR00379935
ENGR00372160	NewWork	[FLS] Optimize precompile parameter as define 'NewWork Description: Because Autosar parameter "FlsPageSize" is a precompile parameter and also platform specific with a constant value, it is more efficient to use it as a define instead of an array in the configuration structure. Requirement source: NA Proposed solution: Move "PageSize" parameter from the configuration structure to Fls_Cfg.h as a define. Change all calls in the driver and IP code.",'Update driver code for BETA 0.9.0
ENGR00372163	NewWork	[FLS] Optimize precompile parameter as define 'NewWork Description: Because Autosar parameter "FlsPageSize" is a precompile parameter and also platform specific with a constant value, it is more efficient to use it as a define instead of an array in the configuration structure. Proposed solution: Move "PageSize" parameter from the configuration structure to Fls_Cfg.h as a define. Change all calls in the driver and IP code.",'Update driver code for BETA 0.9.0
ENGR00364091	Defect	[FLS] Plugin generation is not aligned with MCAL standard 'Problem detailed description (how to reproduce it): This ticket it's only an internal rework to have all drivers configurations generated in the same way. There is no functional impact. 1. The file Fls.mak contains the generation of the Fls_Cfg.h, Fls_Cfg.c and Fls_PBcfg.c into the plugin directory "generate". The correct one must be "generate_PC". Please update the file Fls.mak to reflect the approach of the MCAL standard 2. The file Fls.mak has a tag "PLG_CUSTOM_GENS" that is empty. It must contain the dirs/variants for which the generation of the plugin exists. For FLS, the correct tag must be "PLG_CUSTOM_GENS = generate_PC,generate_PB" The external impact resumes to a different directory name/path for the configuration files inside the FLS plugin. No other functional impact. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: 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): The files: Fls_Cfg.h, Fls_Cfg.c, Fls_PBcfg.c and Fls_VersionCheck.m should be generated to: "outdir=generate_PC" and "outdir=generate_PB", instead of: "outdir=generate".","'Make Fls Plugin generation aligned with MCAL standard Please refer to description in the main tab for more details
ENGR00378536	NewWork	[FLS] Re-name variables and structure field names to follow sMCAL code rules 'NewWork Description: Arcording to SMCAL coding guidline V4.0 April 2016 SMCALRule 2.23 the global variables follow the naming convention <Msn>_[<Ip>_] [<PrefixType>]<VarName> ? where <PrefixType> might be: - p: pointer to data - pf: pointer to a function - u (for unsigned), s (for signed): integers for which size is not relevant or is unknown (usually HLD related data) - e: enum variables - u8, u16, u32, s8, s16, s32: sized integers, for which size is relevant (usually for the register values) - b: Boolean variables to be compared against TRUE/FALSE - a: arrays - nothing: for anything else Rule 2.23 The local variable names and structure field names following the naming convention [<PrefixType>]<VarName> (where <PrefixType> is defined as for Rule 2.23) ? [Rule 2.24] Note: For configuration structures the field name might be prefixed with <Msn>_[<Ip>_] for being similar with the names used in the plugins But, There is no prefix-type in FLS's variables and structure field names Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Add prefix-type for variables and structure filed names",'Add prefix-type for variables and structure filed names,ENGR00378622
ENGR00374259	Defect	[FLS] Remove "EDITABLE" tag from FlsUseInterrupts .xdm parameter 'Problem detailed description (how to reproduce it): IPV_FLASHV2 IP code implementation does not support interrupt processing. This is stated as a note for the "Fls Use Interrupts" configuration parameter and in the user manual. If "Fls Use Interrupts" configuration parameter is configured as enabled, then compilation issues might appear. It is better that the configuration parameter should not be configurable to enabled when not supported. Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		"FlsUseInterrupts" parameters is enabled on platforms which do not support interrupt processing. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: "FlsUseInterrupts" can be enabled on all platforms. Expected behavior: "FlsUseInterrupts" should only be available to be enabled only on platforms which support interrupts. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Change <a:da name="EDITABLE" value="true"/> with <a:da name="READONLY" value="true"/> in Fls.xdm","'Configuration parameter "Fls Use Interrupts" defined as read-only on platforms which do not support interrupts.
ENGR00366513	NewWork	[FLS] Replace STATIC definition with static (lower case) 'NewWork Description: STATIC definition was removed from Base, FLS should use the C keyword static instead. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Replace STATIC with static (lower case) in fls.c",'Replace STATIC definition with static (lower case)
ENGR00377793	NewWork	[FLS] Rework the generic files to enable the user mode only for the needed platforms 'NewWork Description: Rework the generic files to enable the user mode only for the needed platforms. Use resource parameter to configure if user mode is needed or not. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add a new resource parameter that will be use to configure to make FlsEnableUserModeSupport","'-Add Fls.User.Mode.Support as a new parameter in the resource. It will be used to configure FLS in each platforms will support User Mode or not. - Changed file : Fls resource files, Fls.xdm
ENGR00376353	Defect	[FLS] Sector list reorderd alphabetically when generating the .epc with Tresos

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): The sector list is reordered alphabetically when generating the .epc with Tresos. Because of this, the index used to iterate and verify the SectorStartAddress to be in order becomes invalid. This issue does not appear when generating the .xdm configuration file. This issue becomes problematic only when using more than 10 configured flash sectors. Preconditions: Using the .epc file for configuration, not the .xdm Configure more than 10 flash sectors or Change the name of the flash sectors out of alphabetic order. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The sector list is reordered in .epc files, and instead of: FlsSector_0 FlsSector_1 FlsSector_2 .... we get: FlsSector_0 FlsSector_1 FlsSector_10 FlsSector_11 ... Because of this, if that configuration file is used, at generate the following errors are thrown: ERROR: (1019) Invalid value for node "/AUTOSAR/TOP-LEVEL-PACKAGES/Cfg/ELEMENTS/Fls/FlsConfigSet/FlsConfigSet_0/FlsSectorList/FlsSector/FlsSector_10/FlsSectorStartaddress": Invalid start address of the sector. Hit the calc button to calculate correct value. ERROR: (1019) Invalid value for node "/AUTOSAR/TOP-LEVEL-PACKAGES/Cfg/ELEMENTS/Fls/FlsConfigSet/FlsConfigSet_0/FlsSectorList/FlsSector/FlsSector_2/FlsSectorStartaddress": Invalid start address of the sector. Hit the calc button to calculate correct value. Expected behavior: Create the list in .epc ordered as in Tresos, based on the index. In this case, the list should be: FlsSector_0 FlsSector_1 FlsSector_2 FlsSector_3 ... FlsSector_9 FlsSector_10 FlsSector_11 FlsSector_12 Proposed solution (Optional): Add a new Vendor specific parameter (ex.: FlsSectorIndex) which to use to iterate over the flash sectors. This parameter has to be used instead of the current @index parameter.","'Added a new Vendor specific parameter ("FlsSectorIndex") inside "FlsSector" which is intended to be used, internally, instead of "@index" to iterate over the flash sectors. When creating a configuration in Tresos, the value of "FlsSectorIndex" is intended to be the

Table continues on the next page...

ID	Subtype	Headline and Description
		same as the index value.
ENGR00379671	NewWork	[FLS] Update IM with proposed action for application handler for ECC errors 'NewWork Description: Update description of the Data Cache influence on ECC error exception reporting. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Current ECC management description from IM, states that for asynchronous interrupts, the application machine check handler has to be updated, using the below text: "Note: If data cache is enabled, even if the flash region is configured as cache-inhibited in the memory protection unit, an asynchronous ECC exception might be triggered. In case the ECC exception is asynchronous, triggered by a cache line prefetch, the syndrome and handler return address might be different than the case when the exception was triggered by a direct load instruction. The application machine check handler should be updated accordingly." Delete both paragraphs which threat the cache influence from the "ECC Management on Data Flash" sub chapter, because they do not affect the flash data sectors and add a description to the previous sub chapter, "Implementing an Exception Handler in case of non correctable ECC error", like the one proposed below: " On PowerPC platforms, if data cache is enabled and even if the flash region is configured as cache-inhibited in the memory protection unit, a cache data line fill to the flash memory may be triggered by the driver`s load instruction. The read data will not be marked as valid in the cache, but the cache fetch might trigger an ECC exception if there is an ECC affected location inside the same cache line as the location read by the driver. For example: if the flash driver accesses for read location 0x10000000, and there is an ECC affected location at address 0x10000008, an ECC event might be triggered by a cache line fill. To cover this behavior, the Fls_DsiHandler will report a job as failed if there is any ECC event in the same cache line as the currently accessed memory location. Note:To maintain the robustness level in the FEE driver, the "Fee_VirtualPageSize" should be configured according to the cache line size (32 bytes). If the ECC exception is influenced by a cache fetch, depending on the position of the location read by driver and the position of the ECC affected location inside the cache line, the triggered exception might be synchronous or asynchronous.

Table continues on the next page...

ID	Subtype	Headline and Description
		If the exception is synchronous (LD bit of the syndrome is set) the value reported in MCSRR0 represents the address of the instruction which caused the exception, and should be skipped if return to driver code is needed. If the exception is asynchronous (LD bit of the syndrome is cleared) the value reported in MCSRR0 represents the address of the instruction which would have executed next, had the exception not appeared, thus it would have to be executed. </p>"","Update IM chapter for ECC errors handling.
ENGR00373934	NewWork	[FLS] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.","'Use "READONLY" for those parameter in Fls.xdm FlsTotalSize FlsBaseAddress FlsNumberOfSectors FlsCallCycle
ENGR00378360	NewWork	[FLS] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA 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): [...]",'Fix all MISRA errors that are not in in MISRADeviation.xls,ENGR00379002
ENGR00361366	Defect	[FLS] 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_SP EC_VERSION_PATCH"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJO R"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO R"!][!// [!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: Add version checking for configuration files 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",'Add version checking in configuration template files.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00366650	Defect	[FLS] Wrong in checking condition to report Dem events 'Problem detailed description (how to reproduce it): The name of the define used to guard DEM error report is not consistent with the naming convention used in other MCAL drivers. Flash driver is checking to report Dem event using the following construct: #if (FLS_DEM_ERROR_DETECT == STD_ON) if ((VAR(boolean, FLS_VAR))TRUE == Fls_ConfigPtr->Fls_E_EraseFailedCfg.state) { Dem_ReportErrorStatus((VAR(Dem_EventIdType, AUTOMATIC))Fls_ConfigPtr->Fls_E_EraseFailedCfg.id, DEM_EVENT_STATUS_FAILED); } #endif 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: Improve the naming macro to "FLS_DISABLE_DEM_REPORT_ERROR_STATUS" instead of "FLS_DEM_ERROR_DETECT" (in generation file).","'Flash driver is checking to report Dem event as following: #if (FLS_DEM_ERROR_DETECT == STD_ON) if ((VAR(boolean, FLS_VAR))TRUE == Fls_ConfigPtr->Fls_E_EraseFailedCfg.state) { Dem_ReportErrorStatus((VAR(Dem_EventIdType, AUTOMATIC))Fls_ConfigPtr->Fls_E_EraseFailedCfg.id, DEM_EVENT_STATUS_FAILED); } #endif Just suggest to improve naming maro FLS_DISABLE_DEM_REPORT_ERROR_STATUS instead of FLS_DEM_ERROR_DETECT (in generation file).",ENGR00367264
ENGR00370100	NewWork	[FR] Activate Elektrobit Tresos Fibex import 'N/A,'Activate Fibex Importer
ENGR00372517	Defect	[FR] Add inclusion of Soc_Ips.h in Fr_Cfg.h 'Problem detailed description (how to reproduce it): In FR driver, the Fr_Cfg.h include file Compiler.h so the Fr_Cfg.h also include

Table continues on the next page...

ID	Subtype	Headline and Description
		Soc_ips.h. Therefore, the the defines for errata in Soc_ips.h are exists and and active the errata workaround Due to CR ENGR00364842, Compiler.h shall not include Soc_ips.h. So the driver doesn't run the workaround for errata even if there are errata defines in Soc_ips.h Expected behavior: The driver must enable the errata workaround Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add inclusion of Soc_Ips.h in Fr_Cfg.h","Fr_Cfg.h
ENGR00364496	Defect	[FR] Correct misra violation 'Problem detailed description (how to reproduce it): There are some misra error due to change size of base address and type definition for some platform. Preconditions: Misra report Test Case ID (internal TC that caught the defect) - optional NA Trigger: Check misra Observed behavior: Violation in Misra log and unable to generate misra report Expected behavior: Able to generate misra report Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add proper comment because this is specific for ARM64 platform.","Correct misra comments
ENGR00378685	Defect	[FR] Correct naming for define related to byte order. 'Problem detailed description (how to reproduce it): The naming of the define "FR_HIGH_DATA_BYTE_FIRST" is incorrect. The segment code into this define which use to swap location byte and this swap only need to Little Endian architect. Therefore, this define should be to "FR_LOW_DATA_BYTE_FIRST". This problem will be fix into other CR. Preconditions: Enable this define for Littel Endian platform. Expected behavior: The naming shall be right, describe correctly about the target of code segment depend this define. Proposed solution (Optional): This define should be to "FR_LOW_DATA_BYTE_FIRST".","Fr_FlexRay.h and Fr_FlexRay.c
ENGR00372568	Defect	[FR] Fixing MISRA errors for RaceRunner Ultra 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		Correct or comment Misra errors in the source code. 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):",'Fixed MISRA violation
ENGR00378345	NewWork	[FR] 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Fr_FlexRay.c,ENGR00379109
ENGR00364019	Defect	[FR] Incorrect receive/transmit data when data byte order is reversed 'Problem detailed description (how to reproduce it): In TreeRunner platform (ARM architecture), for each 2-byte block in message buffer data field, the payload data byte which has higher index is located lower address. The payload data bytes are reversed in 2-byte block when receive or transmit Observed behavior: When transmit/receive data Expected behavior: Transmit/receive data correctly when data byte order is reversed Proposed solution (Optional): Check the data byte order in message buffer data fields Implement receive/transmit payload data base on the data byte order in message buffer data fields",'Fr_Cfg.h,ENGR00364028
ENGR00373906	NewWork	[FR] Investigate and implement to reduce cyclomatic complexity and nesting level 'Requirements: - 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. 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 ---------- Implementation details: In AUTOSAR_MCAL_<MDL>_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments - for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment

Table continues on the next page...

ID	Subtype	Headline and Description
		- for all the nesting depth >4 the problems shall be resolved",'separate function in order to reduce cyclomatic level,ENGR00380833
ENGR00363268	NewWork	[FR] Issue while casting address to uint32 in 64-bit platform 'NewWork Description: In 64 bits platform, casting an address to uint32 cause issue. The compiler does not allow to do so. Therefore, please change the address from uint32 to uint64 for 64-bit platform. Expected behavior: Driver work with 64 bits microcontroller. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Using different line of code for 32 and 64 bits platfrom.",'Update config structure to store address as uint64 instead of uint32.
ENGR00376096	Defect	[FR] Redundant frame ID checking in dynamic segment in Fr_Flexray_ReconfigLPdu 'Problem detailed description (how to reproduce it): In Fr_Flexray_ReconfigLPdu line 4968, the code as below: if(frameId > (Fr_pCfgPtr + ctrlIdx)->LowLevelConfigSetPtr->gNumberOfStaticSlots) { u16tmpRegVal_37 |= FLEXRAY_MBCCFR_CHA_U16; /* Clear CHB bit */ u16tmpRegVal_37 &= ~FLEXRAY_MBCCFR_CHB_U16; } it does not need to check frame ID > gNumberOfStaticSlots again because this condition has been checked in Fr_ReconfigLPdu before. Proposed solution (Optional): Remove frame ID checking in dynamic segment",'Fr_FlexRay.c
ENGR00368641	Defect	[FR] Remove the dependence from Resource in generate template file 'Problem detailed description (how to reproduce it): In generate template file, we currently select information in Resource module to generate line. For example: [!SELECT "as:modconf('Resource')[1]"![!// [!IF "node:exists(ResourceGeneral/ARM_CoreArchitecture)"![!// [!IF "(ResourceGeneral/ARM_CoreArchitecture = 'ARM_A64_ARCH')"![!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint64)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ELSE!][!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint32)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ENDIF!][!// [!ELSE!][!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint32)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0])

Table continues on the next page...

ID	Subtype	Headline and Description
		[!ENDIF!] [!ENDSELECT!][!// It's correct for TreeRunner platform, which use Resource epc file when generating and building source code. For other platforms, which not use Resource.epc so it template file does not generate the line Preconditions: Orther platform except TreeRunner Test Case ID (internal TC that caught the defect) - optional [...] Trigger: In generate time 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): Add a variable to check if generate file use the Resource before generate the line. Please see the example source code: [!VAR "ResourceExist" = "'false'"!] [!SELECT "as:modconf('Resource')[1]"!][!// [!"$ResourceExist = 'true'"!][!// [!IF "node:exists(ResourceGeneral/ARM_CoreArchitecture)"!][!// [!IF "(ResourceGeneral/ARM_CoreArchitecture = 'ARM_A64_ARCH')"!][!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint64)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ELSE!][!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint32)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ENDIF!][!// [!ELSE!][!// [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint32)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ENDIF!][!// [!ENDSELECT!][!// [!IF "$ResourceExist = 'false'"!] [!WS "4"!]#define FR_CC[!"$CtrlIndex"!]_FLEXRAY_MEMORY_BASE_ADDR ((uint32)&Fr_MemoryArea_Ctrl[!"$CtrlIndex"!][0]) [!ENDIF!][!//",'Fr_Cfg.h
ENGR00366636	NewWork	[FR] Update user manual in order to describe LPDU configuration in order to allow message buffer sharing 'NewWork Description: Update user manual in order to include a detailed remark which shall describe the conditions under which the message buffer sharing algorithm is usable. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): The configuration tool(Tresos) tries to share one physical resource (i.e.

Table continues on the next page...

ID	Subtype	Headline and Description
		message buffer) for several LPdu's (FlexRay frames) if the following conditions are met: +Buffer reconfiguration functionality is enabled for FlexRay driver in FrGeneral/ VendorSpecific container. +Given LPdu's are not assigned to any FIFO storage +Given LPdu's are configured for operation in static segment of communication cycle +PREPARE_LPDU communication actions are defined for these LPdu's in FrIf job list For each LPDU, the configuration tool calculates timing intervals during which a hardware MB is needed to do a transmission or a reception. The time is expressed in macroticks and is calculated over a single time axis which covers all 64 cycles. The start and end times of each timing interval are calculated as follows: The start time is defined always by the time value of the PREPARE_LPDU action. The end time is defined for: 1) Transmission activities by: ? either time of TX_CONFIRMATION action ? or time of physical data transmission on the bus indicated in FrameTimeTrigger which is referred by the LPDU. 2) Reception activities by: ? either time of RECEIVE_AND_INDICATE action ? or time of RECEIVE_AND_STORE action ? or RX_CONFIRMATION action If two LPDUs has each at least one MB usage interval for which the range values overlaps a little bit, they cannot share a physical message buffer. The sharing algorithm compare each LPDU with all other LPDUs in order to identify a LPDU pair which can share the same MB.",'UM
ENGR00373397	NewWork	[FR] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Conclusion: 1) Nodes that have constant false editable conditions (<a:a name="EDITABLE" value="false"/>) will change to (<a:a name="READONLY" value="true"/>)

Table continues on the next page...

ID	Subtype	Headline and Description
		2) Exception: the 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/> 3) Nodes that have EDITABLE attributes with type xPath will not be changed to READONLY",'Fr.xdm
ENGR00378364	NewWork	[FR] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Update code in order to fix misra errors,ENGR00380428
ENGR00370993	Defect	[FR] Wrong generation if FIFO range contains ID of Tx Frame 'Problem detailed description (how to reproduce it): Please get the details from email in attachment field. Suggest: If Rx FIFO range contain ID of Tx message buffer 1. Warning message in EB tresos generation interface. (user need to know "Ignore Rx Fifo") 2. Ignore Rx Fifo.",'N/A
ENGR00381374	Defect	[FR]Fix misra error. 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate some rule 5.1, 17.4, rule 11.3 and 13.7. When issue in tool is fixed, we have to fix or comment the violations related to these rules. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Have error messages prevent generating Misra report Expected behavior: Can generate Misra report without any error 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): Fix the violation if possible or comment the violations with deviation reasons",'up date code to fix misra error,ENGR00381376
ENGR00380783	Defect	[Fee] The legacy OFF mode is allowed even if no immediate data is configured 'Problem detailed description (how to reproduce it): Legacy OFF is intended for reserving an area for immediate blocks Legacy ON is intended if you don't need to reserve area for immediate blocks So having the legacy ON mode with no immediate block should not be allowed by the tool. Preconditions: Fee has configuration -FEE_LEGACY_MODE STD_OFF -no Fee block is an immediate block Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The configurator allows configuration: -FEE_LEGACY_MODE STD_OFF -no Fee block is an immediate block Expected behavior: The configurator should issue error for configuration: -FEE_LEGACY_MODE STD_OFF -no Fee block is an immediate block Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Change Fee xdm or configuration template to issue error for configuration: -FEE_LEGACY_MODE STD_OFF -no Fee block is an immediate block",'- Update Fee configuration template code to throw an error message when FeeLegacyMode is off and there is no immediate block in the FeeBlockConfiguration
ENGR00370116	NewWork	[GPT, ICU, MCU, OCU, PWM][MCL] Update Base address array to display all HW Units 'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR

Table continues on the next page...

ID	Subtype	Headline and Description
		DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the adress 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....","'Update Base address array to display all HW Units File modified: -eMios_Common.c
ENGR00362313	NewWork	[GPT] Add PIT_1 support for Treerunner 'NewWork Description: New GPT channels must be added to the IP (PIT_1) for Treerunner. Requirement source: S32_V234_RM_Rev1_DraftE_PartI_review.pdf (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Add PIT_1 support for platforms that have more then 1 PIT modules.,ENGR00366319
ENGR00371071	Defect	[GPT] Changing mode from GPI to MCB could cause a interrupt 'Problem detailed description (how to reproduce it): Changing mode from GPI to MCB can cause a spurious interrupt Preconditions: eMios Ch8 is used to generate Pwm signal, and Ch14 is used in the Gpt driver as one shot mode, with an internal counter. Test Case ID (internal TC that caught the defect) - optional NA Trigger:

Table continues on the next page...

ID	Subtype	Headline and Description
		Having Ch8 as Pwm generator Observed behavior: Interrupt sometime occured immediately after changing to MCB mode (when ussing Gpt_StartTimer) 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): Follow the following steps in Gpt_eMios_StartTimer: - Save interrupt enable bit - Clear interrupt enable bit - Change the mode from GPI to MCB - Clear the interrupt flag - Write the interrupt enable bit, with the saved value",'Fixed hardware issue with spurious interrupts when chaning from GPI to MCB mode.
ENGR00366619	NewWork	[GPT] Define a variable that determines to use a single interrupt or separate interrupts in PIT 'NewWork Description: Each PIT module (PIT_0, PIT_1,...) uses either a single interrupt or separate interrupt sources for each channels depending on specific hardware. So, defining a variable that determines to use a single interrupt or separate interrupts in PIT. Proposed solution (Optional): Defines GPT_PIT_MODULE_SINGLE_INTERRUPT variable and uses it to check which parts of code will handle PIT's ISR. For example: #if (GPT_PIT_MODULE_SINGLE_INTERRUPT == STD_ON) #ifdef GPT_PIT_0_ISR_USED ISR(Gpt_PIT_0_ISR); #endif #else #ifdef GPT_PIT_0_CH_0_ISR_USED ISR(Gpt_PIT_0_Ch_0_ISR); #endif #endif /*GPT_PIT_MODULE_SINGLE_INTERRUPT == STD_ON*/","'Files modified: Gpt_Pit.h
ENGR00366614	NewWork	[GPT] Define a variable that determines to use a single interrupt or separate interrupts in STM 'NewWork Description: Each STM module (STM_0, STM_1,...) uses either a single interrupt or separate interrupt sources for each channels depending on specific hardware. So, defining a variable that determines to use a single interrupt or separate interrupts in STM. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Defines GPT_STM_MODULE_SINGLE_INTERRUPT variable and uses it to

Table continues on the next page...

ID	Subtype	Headline and Description
		check which parts of code will handle STM's ISR. For example: #if (GPT_STM_MODULE_SINGLE_INTERRUPT == STD_ON) #ifdef GPT_STM_0_ISR_USED ISR(Gpt_STM_0_ISR); #endif #else #ifdef GPT_STM_0_CH_0_ISR_USED ISR(Gpt_STM_0_Ch_0_ISR); #endif #endif /*GPT_STM_MODULE_SINGLE_INTERRUPT == STD_ON*/",'Gpt_Stm.c
ENGR00381398	Defect	[GPT] Dependency at GptFreezeEnable between PIT/STM channels as unnecessary 'Problem detailed description (how to reproduce it): The user should be able to select different values for the GptFreezeEnable checkbox on each STM/PIT module. Right now, there is a error in Tresos for such a case. Only for the same module there must be the same value of the checkbox. Preconditions: User wants to configure the freeze bit, independently on each STM/PIT module Test Case ID (internal TC that caught the defect) - optional n/a Trigger: generating Observed behavior: An error message when selecting different values for GptFreezeEnable on different STM/PIT modules. Expected behavior: Without error. 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",'Files modified: Gpt_Cfg.h
ENGR00375637	Defect	[GPT] EMIOS_x_ISR_USED defines should be only generated for channels which use notification 'Problem detailed description (how to reproduce it): At the moment, EMIOS_x_ISR_USED are generated for all cases, if a GPT channel appears, disregarding the actual use of ISR's. This might lead to unnecessary issues in checks when interrupts are shared: #if ((defined ICU_EMIOS_0_CH_1_ISR_USED) && (defined GPT_EMIOS_0_CH_1_ISR_USED)) #error "ICU and GPT resource conflict for EMIOS unified channel EMIOS_0_1" #endif Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Whenever an EMIOS channel is configured, no matter the channel configures

Table continues on the next page...

ID	Subtype	Headline and Description
		notification or not, EMIOS_x_ISR_USED define is always generated. Expected behavior: EMIOS_x_ISR_USED define should only be generated when the channel configures notification. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Adding check whether an EMIOS channel configures notification, and then, EMIOS_x_ISR_USED define will be generated.','Gpt_Cfg.h
ENGR00372541	Defect	[GPT] Fix MISRA errors in Gpt.c 'Problem detailed description (how to reproduce it): Value is returned from Det_ReportError function hasn't been used anywhere which caused a MISRA violation Rule 16.10. Preconditions: Change Next TimeOut mode is configured to used. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Misra rule 16.10 violation. 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): Commenting MISRA warning 16.10 from calling Det_ReportError. .','Gpt.c
ENGR00373203	Defect	[GPT] Fix compiler warnings for PIT and STM's ProcessCommonInterrupt 'Problem detailed description (how to reproduce it): warning #177-D: function "Gpt_Pit_ProcessCommonInterrupt" was declared but never@94 warning #177-D: function "Gpt_Stm_ProcessCommonInterrupt" was declared but never@95 Preconditions: When PIT or STM channels are configured to use but do not use their ISR. Test Case ID (internal TC that caught the defect) - optional Trigger: Observed behavior: Expected behavior: These functions (Gpt_Pit_ProcessCommonInterrupt and Gpt_Stm_ProcessCommonInterrupt) should be disable whenever there is no any 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):","'Files modified: Gpt_Pit.c Gpt_Cfg.h",ENGR00374291

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00381452	NewWork	[GPT] Fixed code review against checklist for Calypso 3M/6M RTM 1.0.2 'NewWork Description: Code Review review will done using this CR; the result of this activity will be the code review checklist, this CR is treated as "platform specific" New CR will be raised for the code modifications resulted for the "code review" activity. This CR will be used to update code, will be analyzed for all platforms and changes will be integrated for all affected platforms Requirement source: sMCAL Release criteria document: http://compass.freescale.net/go/228798570 Coding guideline: http://compass.freescale.net/go/227962899 Code review checklist: http://compass.freescale.net/go/224108405 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A","'File modified: Gpt_Rtc.c Gpt_Rtc_Type.h",ENGR00381453 ENGR00381454 ENGR00381455
ENGR00370892	Defect	[GPT] GPT driver cannot support configuring the source clock of STM timer 'Problem detailed description (how to reproduce it): I used STM_0 as the wakeup sources to wake up MCU module from STANDBY mode. In RM, The STM timer can choose Sys_Clk or FXOSC as source clock. But currently, GPT driver didn't support to configure for source clock of STM timer, so the Sys_Clk is always default source clock of STM timer (we cannot select FXOSC as source clock for STM). In STANDBY mode, Sys_Clk is not available but FXOSC is available. When MCU driver takes the chip to STANDBY mode, STM timer is disconnected with source clock (Sys_Clk), so MCU can't wakeup the chip from STANDBY mode by STM. Alin Meleandra [b46093] - This issue is present only on Calypo 3M (MPC5746C) variant. No other uC using STM IP appears to have the CSL bit. Such small change is quite hard to identify when porting for new platform starts. Preconditions: n/A Test Case ID (internal TC that caught the defect) - optional n/a Trigger: N/A Observed behavior: Did not support source clock selection for STM channels in EB Expected behavior: Gpt driver should supports to select source clock for STM channels 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","'Files modified: Gpt_Ipw_Types.h Gpt_Cfg.h Gpt_PluginMacros.m Gpt_Ipw.c Gpt.xdm",ENGR00371330

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00378348	NewWork	[GPT] 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 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: AUTOSAR_MCAL_GPT_EXCLUSIVE_AREAS.xls

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",'Files Modified: Gpt.c,ENGR00378843 ENGR00378844
ENGR00379318	NewWork	[GPT] Implement findings from code review against checklist for Matterhorn RTM 1.0.1 'o Review code review against checklist o Implement findings Checklist is attached to the umbrella ticket and available at: http://compass.freescale.net/livelink/livelink? func=ll&objId=224108405&objAction=browse","'Gpt.c Gpt_Cfg.h Gpt_Cfg.c Gpt_PBcfg.c",ENGR00379319 ENGR00379320
ENGR00372898	NewWork	[GPT] Improvement for SRAM usage for configured channels. 'NewWork Description: Currently the GPT driver is designed for performance optimization and simplification. However, there are some variables waste ~1000bytes and ~500bytes if only few GPT channels are configured. STATIC VAR(Gpt_ChannelInfoType, GPT_VAR) Gpt_aChannelInfo[GPT_HW_CHANNEL_NUM] volatile STATIC VAR(Gpt_ValueType,GPT_VAR) Gpt_aStopTime[GPT_HW_CHANNEL_NUM] Pwm_GtmAtom_aChannelNotifType[PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmAtom_aChannelPeriod[PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmTom_aChannelPeriod[PWM_GTM_TOM_HW_CHANNELS_NO] GPT_HW_CHANNEL_NUM, PWM_GTM_ATOM_HW_CHANNELS_NO,PWM_GTM_TOM_HW_CHANNELS_NO are maximum channels supported in the device. However the customer may use only serveral channels (even only 1 channel). Can Gpt driver optimized to recude the SRAM waste? e.g Generate the above macro base on the number of configured channels. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Gpt_Cfg.h
ENGR00380152	NewWork	[GPT] Including Reg_eSys_eMios.h only needed with eMios 'NewWork Description: Dependency between Gpt and Mcl when using eMios, but when using a PIT or STM we don't see any dependency. However if we use PIT or STM we have to add in our project Mcl also and this should not be the case. Reg_eSys_eMios.h file comes from Mcl and it's included in Gpt_PBcfg.c and Gpt_Cfg.c. Maybe a check can be added that if eMios is used, only then include Reg_eSys_eMios.h

Table continues on the next page...

ID	Subtype	Headline and Description
		Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Adding the condition to check if eMios is used => include Reg_eSys_eMios.h else => don't need to include this file. so, with customer project, they don;t need to include MCL into.","'File modify: Gpt_PBcfg.c and Gpt_Cfg.c
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"

Table continues on the next page...

ID	Subtype	Headline and Description
		',`dnl #include "MemMap.h" ')dnl static VAR(uint8, MCU_VAR) Mcu_au8ClockConfigIds[MCU_MAX_CLKCONFIGS];",'Gpt.c,ENGR00368951
ENGR00373416	Defect	[GPT] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in our MCAL drivers. Variable declarations should have the definitions and all extern declarations mapped to the same memmap sections. See attached file for all rules specifying the use of Memmap sections. Observed behavior: Global variables used by Mcal drivers are mapped to default .data sections. Expected behavior: All global variables used by Mcal drivers must be mapped to the special .mcal_data sections. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]","'Files modified: Gpt_Cfg.c Gpt_PBcfg.c
ENGR00379036	Defect	[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):

Table continues on the next page...

ID	Subtype	Headline and Description
		Change GptChannelTickValueMax in Gpt.xdm and Gpt_PluginMacro.m","'Files Modified: Gpt.xdm Gpt_PluginsMacro.m
ENGR00381031	Defect	[GPT] Incorrect calculation of GptChannelTickFrequency for RTC 'Problem detailed description (how to reproduce it): On Calypso, all 4 RTC source clocks are divided by either 32 or 512 prescaler. But currently, with XOSC clock source selected, frequency of RTC channel was divided TWICE by GptChannelPrescaler: (num:i(node:fallback(../GptRtcChannelClkSrc,'RTC_GPT_CLKSRC_XOSC') = 'RTC_GPT_CLKSRC_XOSC')) * (noderef:value(concat(noderef:ref(noderef:value(concat(noderef:ref(node:fallback(../GptChannelClkSrcRef, .)), '/GptClockReference'))), '/McuClockReferencePointFrequency')) div num:i(not(contains(node:fallback(../GptHwChannel,'RTC'),'RTC'))) + node:fallback(../GptChannelPrescaler,1))) div (num:i((num:i(contains((node:fallback(../GptChannelPrescaler,1)),1)) * num:i(1)) + (num:i(contains((node:fallback(../GptChannelPrescaler,1)),2)) * num:i(32)) + (num:i(contains((node:fallback(../GptChannelPrescaler,1)),3)) * num:i(512)) + (num:i(contains((node:fallback(../GptChannelPrescaler,1)),4)) * num:i(16384))) It is wrong. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: XOSC clock source was divided twice by GptChannelPrescaler. Expected behavior: All 4 RTC clock sources are divided by either 32 or 512 prescaler. 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","'Files modified: Gpt.xdm Gpt_PluginMacros.m
ENGR00381028	Defect	[GPT] Incorrect clock selection for STM on Calypso 'Problem detailed description (how to reproduce it): CSL bit in STM_CR register is used to configure clock source for STM channels. But at the moment, the value was written to a wrong register: STM_CCR. /* Select Clock Source */ #if defined(GPT_STM_ENABLECLOCKSWITCH) #if (GPT_STM_ENABLECLOCKSWITCH == STD_ON) /** @violates @ref GPT_STM_C_REF_3 Cast from unsigned long to pointer. */ REG_RMW32(STM_CCR_ADDR32(u8HwChannel), STM_CR_CSL_MASK_U32, (uint32)u8ClockSource<<STM_CR_CSL_SHIFT); #endif #endif

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 Trigger: N/A Observed behavior: STM_CCR register was written to instead of STM_CR register Expected behavior: CSL bit in STM_CR register should be written to . 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",'File modified: Gpt_Stm.c
ENGR00377452	Defect	[GPT] Incorrect usage of Memory Allocation Sections 'Problem detailed description (how to reproduce it): A local variable should be marked as: VAR(typedef, AUTOMATIC) var_name; Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Some local variables were declared as: CONST(typedef, GPT_CODE) var_name; It's wrong syntax for declaring a local variable. Expected behavior: VAR(typedef, AUTOMATIC) var_name; Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): VAR(typedef, AUTOMATIC) var_name;",'File modified: Gpt_Pit.c
ENGR00377035	Defect	[GPT] LDVAL register should be cleared in Gpt_Pit_DeInit 'Problem detailed description (how to reproduce it): LDVAL register should be cleared in Gpt_Pit_DeInit Preconditions: Gpt_DeInit API is enabled Test Case ID (internal TC that caught the defect) - optional NA Trigger: Gpt_DeInit Observed behavior: PIT_LDVAL is not cleared Expected behavior: PIT_LDVAL should be cleared Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add, in Gpt_DeInit:

Table continues on the next page...

ID	Subtype	Headline and Description
		PIT_SETLOADVALUE_U32((uint32)u8HwChannel, 0UL);",'Gpt_Pit.c
ENGR00364211	Defect	[GPT] Modifying a number of channels for each module 'Problem detailed description (how to reproduce it): The number of channels for IP modules (PIT0, PIT1 for example) is defined as a same value which is the maximum of these modules. But, in general, each IP module should have its own define for the number of channels. Preconditions: GPT_PIT_NUM_CHANNEL is defined as the maximum channels among PIT IPs. But in Treerunner, PIT0 and PIT1 have a different number of channels. Therefore, in general approach, using resource files for specific platform, each HW IP should have a separate define for its number of channels. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Correct define as RM Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Change files PlugsMacros.m, Gpt_Cfg.h, resource files to generate out the different number of channels for each module, and we also modify other files related to the use of these variables.","'Files modified: PlugsMacros.m, Gpt_Cfg.h",ENGR00364360 ENGR00364361
ENGR00375746	NewWork	[GPT] Reduce Cyclomatic complexity 'NewWork Description: There are two metrics to be checked (cyclomatic complexity) and (static path depth) For Cyclomatic Complexity -> Values [10, 20] ? can be commented in the StaticAnalysis file but values above 20 should be fixed by breaking the big function into smaller internal functions; same goes for static path depth but the values when value exceed 4... Proposed solution (Optional): in case of Gpt_init(); Gpt_DeInit(); Gpt_SetMode() - Move the For-Loop and NULL PTR check in Gpt_Ipw_StopPredefTimer() -- this means update the Gpt_Ipw_StopPredefTimer() - Move the For-Loops and NULL_PTR check in Gpt_Ipw_StartPredefTimer() -- this means update the Gpt_Ipw_StartPredefTimer() - This means that Gpt_SetMode() function will just call Gpt_Ipw_StopPredefTimer() and Gpt_Ipw_StartPredefTimer()","'Move FOR-LOOPs in Gpt_Init(), Gpt_DeInit(), Gpt_SetMode from Gpt.c to Gpt_Ipw.h and Gpt_Ipw.c Files Modified: Gpt.c; Gpt_Ipw.h; Gpt_Ipw.c
ENGR00371289	NewWork	[GPT] Remove the Java files from Gpt

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: The Java files will be stored in Base. 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=..",'Add JAR file in Base
ENGR00374236	Defect	[GPT] Shared parameters which are configured for individual channels must be the same 'Problem detailed description (how to reproduce it): FTM and STM modules which share a same counter for individual channels, so shared parameters like prescaler, source clock and freeze functinality should be identical for all channels which are using the same counter. Expected behavior: The parameters should be checked in the driver and show a message to users when there is any conflicting parameter. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Define a macro in Gpt_PluginMacro.m [!//This macros is used to check parameters [!MACRO "CheckParameter" , "ModuleType", "param"!] [!IF "$ModuleType = 'STM'"!] [!VAR "MacMaxModules"="num:i(ecu:list('Gpt.Num_Stm_Hw_Modules'))"!] [!ELSEIF "$ModuleType = 'PIT'"!] [!VAR "MacMaxModules"="num:i(ecu:list('Gpt.Num_Pit_Hw_Modules'))"!] [!ELSEIF "$ModuleType = 'FTM'"!] [!VAR "MacMaxModules"="num:i(ecu:list('Gpt.Num_Ftm_Hw_Modules'))"!] [!ENDIF!][!// [!VAR "MacErrorFlag" = "string('0')"!][!// [!FOR "idx" = "0" TO "num:i($MacMaxModules)"!][!// [!VAR "MacErrorFlag" = "string('0')"!][!// [!FOR "x" = "0" TO "num:i(count(GptChannelConfiguration/*))"!][!// [!SELECT "GptChannelConfiguration/*/GptChannelId[.=$x]/.."!][!// [!IF "contains(GptHwChannel, concat($ModuleType, '_', $idx)) and node:exists($param)"!][!// [!IF "($MacErrorFlag = string('0'))"!][!// [!VAR "MacErrorFlag" = "string(node:value($param))"!][!// [!ELSE!][!// [!IF "($MacErrorFlag = string(node:value($param)))"!][!// [!ELSE!][!// [!ERROR!][!// [!"concat('For the ',$ModuleType, '_', $idx,' module all the channels must have the same value for the ', $param, ' parameter, because this module has one general clock prescaler for all its counter channels.')"!][!// [!ENDERROR!][!// [!ENDIF!][!// [!ENDIF!][!// [!ENDIF!][!//

Table continues on the next page...

ID	Subtype	Headline and Description
		[!ENDSELECT!][!// [!ENDFOR!] [!ENDFOR!] [!ENDMACRO!] Calling the macro for shared parameters: [!CALL "CheckParameter","ModuleType"="string('FTM')", "param"="string('GptChannelTickFrequency')"!][// [!CALL "CheckParameter","ModuleType"="string('FTM')", "param"="string('GptFtmChannelClkSrc')"!][// [!CALL "CheckParameter","ModuleType"="string('STM')", "param"="string('GptChannelTickFrequency')"!][!// [!CALL "CheckParameter","ModuleType"="string('FTM')", "param"="string('GptFtmPrescaler_Alternate')"!][!// [!CALL "CheckParameter","ModuleType"="string('STM')", "param"="string('GptStmPrescaler_Alternate')"!][!//","'Files modified: Gpt_Cfg.h, Gpt_PluginMacro.m
ENGR00373365	NewWork	 [GPT] Some Non-Autosar confiugration parameters should have OPTIONAL attribute 'NewWork Description: Summary the discussion between TL and CE: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). Implementation details The parameters that can be excluded from a specific configuration shall be marked as optional (<a:a name="OPTIONAL" value="true"/>, which results in lower multiplicity =0 and upper multiplicity 1). The AUTOSAR parameters have already defined the lower and the upper multiplicity. This change is needed only for the NON-Autosar parameters. The boolean Non-Autosar parameters that enable / disable Non-Autosar features must not be marked as OPTIONAL - only the additional configuration parameters that are needed only when Non-Autosar features are enabled. All parameters that are under this category shall be marked as OPTIONAL=true in the xdm. The check that the features were correctly configured shall be transferred to the boolean parameter that enables the feature. Example: Node1 - configuration node for Non-Autosar feature that selects between transfer mode interrupt or DMA should not be OPTIONAL Node2 - configuration node that selects Dma channel reference needed when DMA transfer is selected should be OPTIONAL when Node1 selects DMA, there should be added an invalid check for the existence of Node2 Reference point SpiPhyTxDmaChannel will be enabled when SpiPhyUnitAsyncMethod is set up to be DMA the changes: into <v:var name="SpiPhyUnitAsyncMethod" type="ENUMERATION"> [...to be added the check that everything is configured the INVALID tag] <a:da name="INVALID" type="XPath"> <a:tst expr="((node:value(.)='DMA') and (node:exists(../

Table continues on the next page...

ID	Subtype	Headline and Description
		SpiPhyTxDmaChannel))) or not(node:value(.)='DMA')" false="Please enable SpiPhyTxDmaChannel"/> </a:da> <v:ref name="SpiPhyTxDmaChannel" type="CHOICE-REFERENCE"> [...] <a:a name="OPTIONAL" value="true"/> <a:da name="EDITABLE" type="XPath" expr="not(node:value(../SpiPhyUnitAsyncMethod) = 'DMA')"/> ---------- Original request from customer engineer---------- Parameters: - Non-autosar parameters. - Specific parameters under certain conditions. This is my additional details: Customer don't use some reference nodes (DMA with Spi, Clock reference with GPT...). Epc file that exported from EB tresos with no reference value in DMA SPI case. For ex: <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyRxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannelAux</DEFINITION-REF> </ECUC-REFERENCE-VALUE> But in EPD file, these parameters are defined as mandatory with lower/upper multiplicity =1. So they should be have default value because a mandatory parameter cannot be empty. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): 1. The not used parameters _should_not_ be part of the Tresos Export. So please change the xdm-file in that way that the parameters will not be part of the export for the case that they are not used! 2. Lower multi","'File Modified: Gpt.xdm Added OPTIONAL attributes in xdm configuration files for NonAutosar nodes that are not mandatory.
ENGR00371144	NewWork	[GPT] Update Base address array to display all HW Units 'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] =

Table continues on the next page...

ID	Subtype	Headline and Description
		{ #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the adress 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....","'File Modified: Gpt_Pit.c",ENGR00371148
ENGR00378367	NewWork	[GPT] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA 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
		[...]","'Fixing MISRA errors Files modified: Gpt.c",ENGR00379194 ENGR00379341
ENGR00380065	Defect	[GPT][PWM] EMIOS_x_ISR_USED defines should be only generated for channels which use ISRs 'Problem detailed description (how to reproduce it): At the moment, EMIOS_x_ISR_USED are generated for all cases, if a GPT channel appears, disregarding the actual use of ISR's. This might lead to unnecessary issues in checks when interrupts are shared: #if ((defined ICU_EMIOS_0_CH_1_ISR_USED) && (defined GPT_EMIOS_0_CH_1_ISR_USED)) #error "ICU and GPT resource conflict for EMIOS unified channel EMIOS_0_1" #endif Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Whenever an EMIOS channel is configured, no matter the channel configures notification or not, EMIOS_x_ISR_USED define is always generated. Expected behavior: EMIOS_x_ISR_USED define should only be generated when the channel configures notification. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): 1. In order to avoid resoure conflicts among ICU, GPT and PWM when an EMIOS channel is configured for two or more modules: + GPT_EMIOS_x_USED defines will be generated if an EMIOS channel is configured to use in Gpt + PWM_EMIOS_x_USED defines will be generated if an EMIOS channel is configured to use in Pwm Then, the resoure conflicts check should be: #if ((defined ICU_EMIOS_0_CH_1_ISR_USED) && (defined GPT_EMIOS_0_CH_1_USED)) #error "ICU and GPT resource conflict for EMIOS unified channel EMIOS_0_1" #endif 2. GPT_EMIOS_x_ISR_USED defines should only be generated if the eMIOS channel configures ISR's in Gpt 2. PWM_EMIOS_x_ISR_USED defines should only be generated if the eMIOS channel configures ISR's in Pwm",'eMios_Common.c
ENGR00368684	NewWork	[ICU] Add checks in Icu_PluginMacros.m 'NewWork Description: Checks and error messages should be added in Icu_PluginMacros.m in order to aid the customer in configuring the driver. Expected behavior: * An error should be added to the customer in case he doesn't configure a Master Bus for channels that are selected to use a master bus source. * Selecting a wrong eMios Master Bus for the configured channel also doesn't give a suggestive error.

Table continues on the next page...

ID	Subtype	Headline and Description
		In our current implementation there's the message: "No value for xpath: $PrescalerVal" Requirement source: Internal (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add checks in Icu_PluginMacros.m",'Adding checks configuration master bus into Icu_PluginMacros.m file.
ENGR00376100	NewWork	[ICU] Channel Idle state should not be handled in Timestamp mode 'NewWork Description: Currently we have Icu_SetBitChState(Channel, ICU_CHANNEL_STATE_IDLE); In the Icu_Timestamp interrupt. This is not needed, because the ACTIVE/IDLE state of the channel is usefull only in Signal Edge Detect and Signal Measurement modes, according to AutoSAR. Idle bit should not be handled in Timestamp mode. Requirement source: ASR. 4.0 / 4.2 - SWS_Icu_00030. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Remove line: Icu_SetBitChState(Channel, ICU_CHANNEL_STATE_IDLE); from Icu_Timestamp","'File modified: Icu.c
ENGR00379031	Defect	[ICU] DMA configuration is not correct for ICU signal measure channel in IPWM/ IPM mode. 'Problem detailed description (how to reproduce it): In Icu_StartSignalMeasurement function, the DMA configuration it only read A2 on 4 trigger and store into the buffer. /* initialize the members of the structure */ tcd_config[Channel].saddr = Icu_Ipw_GetStartAddress \ ((*(Icu_pCfgPtr->Icu_ChannelId))[Channel]); /* @violates @ref Icu_c_REF_7 Violates MISRA 2004 Rule 11.1, Cast from unsigned long * to pointer.*/ /* Compiler_Warning: This warning is thrown because of a conversion between a Icu_ValyeType pointer and a uint32 */ tcd_config[Channel].daddr = (uint32)&Icu_aDmaBuffer[Channel][0]; tcd_config[Channel].ssize = (uint32)ICU_DMA_SIZE; tcd_config[Channel].dsize = (uint32)ICU_DMA_SIZE; tcd_config[Channel].soff = (uint32)0; tcd_config[Channel].doff = (uint32)ICU_DMA_OFFSET; tcd_config[Channel].smod = (uint32)0; tcd_config[Channel].dmod = (uint32)ICU_DMA_MAJORLOOP_COUNT; tcd_config[Channel].num_bytes = ICU_DMA_NUM_BYTES; tcd_config[Channel].iter = ICU_DMA_MAJORLOOP_COUNT; ICU_DMA_SIZE = 4 ICU_DMA_OFFSET = 4

Table continues on the next page...

ID	Subtype	Headline and Description
		ICU_DMA_NUM_BYTES = 4 ICU_DMA_MAJORLOOP_COUNT = 4 This configuration will copy the data in EMIOSA to the buffer on FLAG=1, and this repeats four time to Icu_aDmaBuffer[0]->Icu_aDmaBuffer[3] However, For different modes, the configuration should be different. For SAIC mode, with current implementation, EDSEL=1 (as Activation = ICU_BOTH_EDGES;). The problem is that the starting edge is unknown since the FLAG is set or DMA is triggered on both edge and the A2 is copied to the buffer. For IPWM mode, The A2 and B1 should be copied to the buffer. But the current setting only A2 is copied to the buffer. For IPW mode, with the current implementation, the only A2 is copied to the buffer so the period is calculated from two continous captured values. eg: Period= Icu_aDmaBuffer[1]-Icu_aDmaBuffer[0], Or If DMA configuration is changed so that both A2 and B1 are copied to the buffer and the Period=Icu_aDmaBuffer[1]-Icu_aDmaBuffer[0] after one FLAG=1 even (A2 and B1 are both copied to the buffer). Curretnly the customer would like to have the fix on for ICU with DMA and eMIOS channel in IPWM mode to measure duty and period. Preconditions: eMios channel is configured. Test Case ID (internal TC that caught the defect) - optional Icu_TS_032 Trigger: NA Observed behavior: DMA is not configured for IPMW mode correctly. Expected behavior: DMA is configured correctly for each mode. e.g: A2 and B1 are copied to the buffer correctly. After two events (leading edge and trailing edge) are captured, the DMA compele interrupt occurs to calculate the duty, period and notify to user as well. 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 source code in SAIC mode with DMA feature for SignalMeasurement mode. 2. Remove IPWM/IPM modes (Signal Measurement) with DMA - rationale: DMA is used to reduce the number of interrupts (from 2 to 1 per period in SAIC mode). For IPWM/IPM modes, the number of interrupts is already at 1 per period.","'Allow DMA for SignalMeasurement only in SAIC mode. Fix issues reported by customer.
ENGR00378030	Defect	[ICU] Expand wkpu channel type from uint8 to uint16 'Problem detailed description (how to reproduce it): On some platforms we can have 3 use cases that exceed 255 channels: EG: FTM with logical index from 0 to 7 (ICU_FTM_0_CH_0 -> ICU_FTM_0_CH_7) PORT_CI with logical index from 8 to 392 (ICU_PORT_0_CH_0 -> ICU_PORT_11_CH_31) and WKPU will be from 393 to 424 (ICU_WKPU_CH_0 -> ICU_WKPU_CH_31) In the wkpu case, it's greater than 255 because wkpu is 8 bits type (typedef uint8 Icu_Wkpu_ChannelType) at line 179 in Icu_Wkpu_Types.h file.

Table continues on the next page...

ID	Subtype	Headline and Description
		Therefore one warning that "large integer implicitly truncated to unsigned type " will be generated from compiler whenever parse build. To avoid this one, Icu_Wkpu_ChannelType should be changed to uint16 look like Icu_PORT_CI_ChannelType. Preconditions: Wkpu IP is used Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Warning large integer implicitly truncated to unsigned type Expected behavior: No any warning from compiler. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): typedef uint8 Icu_Wkpu_ChannelType should be replaced by typedef uint16 Icu_Wkpu_ChannelType in Icu_Wkpu_Types.h file.","'File modified: Icu_Wkpu_Types.h
ENGR00379938	Defect	[ICU] Fix issues detected during code review against checklist 'Problem detailed description (how to reproduce it): Fix issued detected during code review against checklist. The eMios IPVault files shall follow the rules of coding convention as bellow: Rule 2.16: Are the data types following the naming convention <Msn>_<TypeName>Type? Is the <TypeName> following the so called CamelCase convention (first letter of each word is uppercase, consequent letters are lowercase) Rule 2.24 "Are the local variable names and structure field names following the naming convention [<PrefixType>]<VarName>" where <PrefixType> might be: - p: pointer to data - pf: pointer to a function - u (for unsigned), s (for signed): integers for which size is not relevant or is unknown (usually HLD related data) - e: enum variables - u8, u16, u32, s8, s16, s32: sized integers, for which size is relevant (usually for the register values) - b: Boolean variables to be compared against TRUE/FALSE - a: arrays - nothing: for anything else and this is more details: Icu_eMios.c Icu_eMios_ChConfigType does not follow rule 2.16 Icu_eMios.c variable local in the function Icu_eMios_GetMasterBus() does not follow rule 2.24 Icu_eMios.c variable local in the function Icu_eMios_DeInit() does not follow rule 2.24 Icu_eMios.c "Parameter of function don't follow rule 2.24: - Icu_eMios_SetPrescaler() - Icu_eMios_SetChConfig() - Icu_eMios_ClearChConfig() Icu_eMios_Types.h Icu_eMios_ChannelType does not follow rule 2.16 Preconditions: None.

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional None. Trigger: None. Observed behavior: Icu_eMios.c and Icu_eMios_Types.h files are violating the coding convention. Expected behavior: Source code should be updated follow coding convention. Proposed solution (Optional): Update Icu_eMios.c and Icu_eMios_Types.h","'Fix code review checklist againt. File modified: Icu_eMios.c
ENGR00378761	Defect	[ICU] Fix issues from design review against checklist 'Problem detailed description (how to reproduce it): In the "ICU File Structure" did not define naming file consistent with structure file in ICU module. The IPW file used in the File structure diagram are pointing to IP Wrapper specific file. (i.e. for Cobra55 the IPW files used should be the ones from ICU_IPW_COBRA55. Also there are discrepancies between the diagram and the inclusion list in the source-code. Preconditions: none. Test Case ID (internal TC that caught the defect) - optional none Trigger: none. Observed behavior: Invalid diagram showing the file-inclusion. Expected behavior: File-inclusion diagram should match the expected one. Proposed solution (Optional): Update design.",'Fix issues from design review against checklist.
ENGR00378349	NewWork	[ICU] Implement data consistency mechanism (exclusive areas) 'NewWork Description: eview 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 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:

Table continues on the next page...

ID	Subtype	Headline and Description
		- 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 Requirement source: - Customer Request - AUTOSAR_MCAL_<DRIVER>_EXCLUSIVE_AREAS.xls - Exclusive areas defined in driver code (all of *.c *.h related files). - Updated IM chapter 5.1 according to exclusive areas defined in the code (IM.pdf) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Implement data consistency mechanism followed up the attached Icu_ExclusiveAreaAnalysis.xlsx file",'Review the implementation of data consistency mechanism (exclusive areas),ENGR00378856 ENGR00379928 ENGR00380834
ENGR00368164	NewWork	[ICU] 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. Requirement source: RM.pdf Proposed solution (Optional): Improve Dem/Det error description in Usermanual",'Improve Dem/Det error description in Usermanual

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00371623	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 the ICU driver: In Icu_Siul2.c, constants should be allocated to the ICU_CONST data sector, so the memclass should be changed from "AUTOMATIC" to "ICU_CONST". Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Issues with correctly alocating the ICU data to the correct memory sector. Expected behavior: All global variables used by Siul2 IPVault must be mapped to the special .mcal_data sections. All constants from SIUL2 IP must have the memclass <MSN>_CONST, in this case, ICU_CONST. Currently they have "AUTOMATIC". Proposed solution (Optional): N/A","'File modified: Icu_Siul2.c",ENGR00372176
ENGR00373417	Defect	[ICU] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in our MCAL drivers. Variable declarations should have the definitions and all extern declarations mapped to the same memmap sections. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Global variables used by Mcal drivers are mapped to default .data sections. Expected behavior: All global variables used by Mcal drivers must be mapped to the special .mcal_data sections. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update source code to make sure that have not any Inconsistent/Incorrect usage of Memory Allocation Sections found","'File modified: Icu_Irq.h Icu.c",ENGR00380830
ENGR00365669	Defect	[ICU] Incorrect version checking in configuration template files

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): 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 Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Have not any of *.m file are tagged above be appearance in specific/generate folder of the driver. Expected behavior: All of *.m file are tagged above have to appear in specific/generate folder of 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): in order to implement all of the following topics that apply : 1. Replace: - ArMajorVersion with ArReleaseMajorVersion - ArMinorVersion with ArReleaseMinorVersion - ArPatchVersion with ArReleaseRevisionVersion 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","'Files modify: -Icu_Cfg.h -Icu_Cfg.c -Icu_PBcfg.c -Icu.mak Adding more files: - Icu_VersionCheck_Inc.m - Icu_VersionCheck_Src.m - Icu_VersionCheck_Src_PB.m
ENGR00381243	Defect	[ICU] Missing the checking condition of IcuMeasurementMode node in Icu.xdm file 'Problem detailed description (how to reproduce it): Currently, In this platform (Calypso) some IPV ex. SIUL2, WKPU do not support for Edge Count, TimeStamp and SignalMeasurement but if user configure IcuHwIP are SIUL2 or WKPU and choose IcuMeasurementMode are ICU_MODE_SIGNAL_MEASUREMENT,ICU_MODE_TIMESTAMP or ICU_MODE_EDGE_COUNTER than all of them are legal and have not any error message to user. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		NA Trigger: NA Observed behavior: Have not any error message inform to user. Expected behavior: An error message should be raised to inform to user 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): Update check the condition into Icu.xdm file for IcuMeasurementMode node.","'File modified: Icu.xdm
ENGR00379721	NewWork	[ICU] Perform code review against checklist for Calypso 3M/6M RTM 1.0.2 'NewWork Description: Code Review review will done using this CR; the result of this activity will be the code review checklist, this CR is treated as "platform specific" New CR will be raised for the code modifications resulted for the "code review" activity. This CR will be used to update code, will be analyzed for all platforms and changes will be integrated for all affected platforms Requirement source: sMCAL Release criteria document: http://compass.freescale.net/go/228798570 Coding guideline: http://compass.freescale.net/go/227962899 Code review checklist: http://compass.freescale.net/go/224108405 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Perform code review and implement against checklist for Calypso 3M/6M RTM 1.0.2,ENGR00380906 ENGR00380907
ENGR00372704	Defect	[ICU] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. It will be replaced by M4_SRC_COPYRIGHTED_TO instead of hard-coded "Freescale Inc". Preconditions: User uses the ICU driver. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Icu_PluginCheck.m and Icu_PluginMacros.m dont have the defines correctly generated. Expected behavior: Copyright information will be filled from the build environment. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): About Copyright should be filled by macro from the build environment.","'Replace hard-coded "Freescale Semiconductor Inc. &

Table continues on the next page...

ID	Subtype	Headline and Description
		STMicroelectronics" to M4_SRC_COPYRIGHTED_TO Files Modified: Icu_PluginCheck.m Icu_PluginMacros.m
ENGR00373366	NewWork	[ICU] Some Non-Autosar confiugration parameters should have OPTIONAL attribute 'Summary the discussion between TL and CE: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). NewWork Description: Implementation details The parameters that can be excluded from a specific configuration shall be marked as optional (<a:a name="OPTIONAL" value="true"/>, which results in lower multiplicity =0 and upper multiplicity 1). The AUTOSAR parameters have already defined the lower and the upper multiplicity. This change is needed only for the NON-Autosar parameters. The boolean Non-Autosar parameters that enable / disable Non-Autosar features must not be marked as OPTIONAL - only the additional configuration parameters that are needed only when Non-Autosar features are enabled. All parameters that are under this category shall be marked as OPTIONAL=true in the xdm. The check that the features were correctly configured shall be transferred to the boolean parameter that enables the feature. Example: Node1 - configuration node for Non-Autosar feature that selects between transfer mode interrupt or DMA should not be OPTIONAL Node2 - configuration node that selects Dma channel reference needed when DMA transfer is selected should be OPTIONAL when Node1 selects DMA, there should be added an invalid check for the existence of Node2 Reference point SpiPhyTxDmaChannel will be enabled when SpiPhyUnitAsyncMethod is set up to be DMA the changes: into <v:var name="SpiPhyUnitAsyncMethod" type="ENUMERATION"> [...to be added the check that everything is configured the INVALID tag] <a:da name="INVALID" type="XPath"> <a:tst expr="((node:value(.)='DMA') and (node:exists(../SpiPhyTxDmaChannel))) or not(node:value(.)='DMA')" false="Please enable SpiPhyTxDmaChannel"/> </a:da> <v:ref name="SpiPhyTxDmaChannel" type="CHOICE-REFERENCE"> [...] <a:a name="OPTIONAL" value="true"/> <a:da name="EDITABLE" type="XPath" expr="not(node:value(../SpiPhyUnitAsyncMethod) = 'DMA')"/> ---------- Original request from customer engineer---------- Parameters: - Non-autosar parameters. - Specific parameters under certain conditions.

Table continues on the next page...

ID	Subtype	Headline and Description
		This is my additional details: Customer don't use some reference nodes (DMA with Spi, Clock reference with GPT...). Epc file that exported from EB tresos with no reference value in DMA SPI case. For ex: <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyRxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannelAux</DEFINITION-REF> </ECUC-REFERENCE-VALUE> But in EPD file, these parameters are defined as mandatory with lower/upper multiplicity =1. So they should be have default value because a mandatory parameter cannot be empty. Proposed solution: 1. The not used parameters _should_not_ be part of the Tresos Export. So please change the xdm-file in that way that the parameters will not be part of the export for the case that they are not used! 2. Lower multiplicity =0 Upper multiplicity =1 for optional. ----------","'File modified: -Icu.xdm
ENGR00372838	Defect	[ICU] The channel state is not updated for Icu signal measurement with DMA 'Problem detailed description (how to reproduce it): Please check following report: I checked the MCAL ICU functions and found that: 1) For non-DMA ICU signal measurement, the function "Icu_SignalMeasurement" will be called in corresponding eMIOS channel ISR (notification function). In this function "Icu_SignalMeasurement", period and duty values are stored, and "Icu_SetBitChState(Channel, ICU_CHANNEL_STATE_IDLE)" is called to set this channel to be IDLE. Thus in "Sample_app_mcal_icu_task.c", the function "Icu_GetInputState()" can get back IDLE state of the specified channel. 2) While for DMA ICU signal measurement, the function "Icu_SignalMeasurementDmaProcessing" is called in corresponding DMA channel ISR notification function. But in this function "Icu_SignalMeasurementDmaProcessing", it only save the period and duty from DMA data buffer, it didn't change the ICU channel state to be "ICU_CHANNEL_STATE_IDLE". So we cannot get correct state of the channel by calling function "Icu_GetInputState()". I added one line of code "Icu_SetBitChState(Channel, ICU_CHANNEL_STATE_IDLE);" at the end of function

Table continues on the next page...

ID	Subtype	Headline and Description
		"Icu_SignalMeasurementDmaProcessing()" and it look like the result is OK. CE'comment: The channel state must be updated when the channel measurement is completed. Preconditions: ICU signal mesaurement with DMA Observed behavior: Icu_GetInputState does not return correct state. Expected behavior: Icu_GetInputState does return correct state. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)","'Update channel state after Signal Measurement with DMA File modified: - Icu.c
ENGR00376119	NewWork	[ICU] The driver should reset the Overflow state in all Start_<Mode> functions 'NewWork Description: Icu_StartTimestamp Icu_StartSignalMeasurement Icu_EnableEdgeCount Should all clear the ICU_CHANNEL_STATE_OVERFLOW bit. ENGR00234546 was the initial implementation of this feature. Expected behavior: The status bit should be explicitly cleared when starting a mode in ICU. Requirement source: Internal refactoring. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)",'TODO
ENGR00379931	Defect	[ICU] The interrupt shall be disabled before changing to GPI mode in Icu_eMios_StopSignalMeasurement and Icu_eMios_StopTimestamp 'Problem detailed description (how to reproduce it): In ICU function "Icu_eMios_StopSignalMeasurement", we can see it first set this eMIOS channel to GPI mode, then disable its interrupt. I think there is potential risk: if this eMIOS channel is set to IPWM mode as ICU channel, and eMIOS interrupt occurred after it's set to GPI mode and before disabling its interrupt. And the ISR will call function "Icu_eMios_SignalMeasurement". This function will calculate period and duty according to current eMIOS channel mode IPWM or SAIC. It should be IPWM mode when calling function "Icu_eMios_SignalMeasurement", but if above special case occurred, the eMIOS channel will be GPI mode when calling function "Icu_eMios_SignalMeasurement". Thus there is small probability to get wrong ICU result. Preconditions: None Test Case ID (internal TC that caught the defect) - optional None Trigger: None Observed behavior: Interrupt is disabled AFTER changing to GPI mode. Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		Interrupt is disabled BEFORE changing to GPI mode. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Rearrange some command lines in source code by move disable interrupt command up before enter GPI mode.","'File modified: Icu_eMios.c
ENGR00373160	Defect	[ICU] The prescaler is not correctly generated for Icu channel 'Problem detailed description (how to reproduce it): The customer reported that The prescaler is not correctly generated for Icu channel Preconditions: Channel: CH_5 Icu Emios Prescaler: EMIOS_PRESCALER_DIVIDE_2 IcuEmiosBusSelect: EMIOS_BUS_DIVERSE Master Bus: Channel: CH_0 Master Bus Prescaler: EMIOS_PRESCALER_DIVIDE_4 as below: Observed behavior: The clock prescaler is not generated correctly for CH_5. CONST(Icu_eMios_ChannelConfigType, ICU_CONST) Icu_eMios_ChannelConfig_PB0[1] = { /** @brief FULL1PM_Icu_EmoiseCh */ { /** @brief eMios HW Module and Channel used by the Icu channel */ ICU_EMIOS_0_CH_5, /** @brief eMios channel configuration parameters */ ((Icu_eMios_ControlType)((Icu_eMios_ControlType) (EMIOS_FREEZE_ENABLE << ICU_EMIOS_FREEZE_PARAM_SHIFT)) | (Icu_eMios_ControlType) ((Icu_eMios_ControlType)EMIOS_PRESCALER_DIVIDE_4 << ICU_EMIOS_PRESCALER_PARAM_SHIFT) | (Icu_eMios_ControlType) ((Icu_eMios_ControlType)EMIOS_DIGITAL_FILTER_BYPASSED << ICU_EMIOS_DIG_FILTER_PARAM_SHIFT) | (Icu_eMios_ControlType)((Icu_eMios_ControlType)EMIOS_BUS_DIVERSE << ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)), (uint8)ICU_BOTH_EDGES, (boolean)FALSE } }; Expected behavior: The clock prescaler is generated correctly for CH_5. CONST(Icu_eMios_ChannelConfigType, ICU_CONST) Icu_eMios_ChannelConfig_PB0[1] = { /** @brief FULL1PM_Icu_EmoiseCh */ { /** @brief eMios HW Module and Channel used by the Icu channel */ ICU_EMIOS_0_CH_5, /** @brief eMios channel configuration parameters */ ((Icu_eMios_ControlType)((Icu_eMios_ControlType)

Table continues on the next page...

ID	Subtype	Headline and Description
		(EMIOS_FREEZE_ENABLE << ICU_EMIOS_FREEZE_PARAM_SHIFT)) | \ (Icu_eMios_ControlType) ((Icu_eMios_ControlType)EMIOS_PRESCALER_DIVIDE_2 << ICU_EMIOS_PRESCALER_PARAM_SHIFT) | (Icu_eMios_ControlType) ((Icu_eMios_ControlType)EMIOS_DIGITAL_FILTER_BYPASSED << ICU_EMIOS_DIG_FILTER_PARAM_SHIFT) | (Icu_eMios_ControlType)((Icu_eMios_ControlType)EMIOS_BUS_DIVERSE << ICU_EMIOS_BUS_SELECT_PARAM_SHIFT)), (uint8)ICU_BOTH_EDGES, (boolean)FALSE } };","'File modified: -Icu.xdm Also update User Manual
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):","'Update after separating the logical and physical configurations. Files modified: -Icu.c -Icu_Ipw.c -Icu_Ipw.h
ENGR00368463	NewWork	[ICU] Update memMap section for local variables '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

Table continues on the next page...

ID	Subtype	Headline and Description
		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"","'File modified: -Icu.c
ENGR00378996	Defect	 [ICU] Update wrong destination address for DMA configuration in signal measurement 'Problem detailed description (how to reproduce it): In Icu.c, the function Icu_SignalMeasurementDmaProcessing updates the destination address and calculate the measured time. /* Compiler_Warning : This warning is thrown because of a conversion between a Icu_ValyeType pointer and a uint32 */ Mcl_DmaUpdateDestAddress(Mcl_DmaChannel, (uint32)Icu_aDmaBuffer[Channel][0]); The destication address shall be address of Icu_aDmaBuffer but not value in Icu_aDmaBuffer[Channel][0]. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The value of first element is configured for DMA destination address. Expected behavior: The address of the buffer (Icu_aDmaBuffer) shall be configured for DMA destination address. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Replace /* Compiler_Warning : This warning is thrown because of a conversion between a Icu_ValyeType pointer and a uint32 */ Mcl_DmaUpdateDestAddress(Mcl_DmaChannel, (uint32)Icu_aDmaBuffer[Channel][0]); by Mcl_DmaUpdateDestAddress(Mcl_DmaChannel, (uint32)&Icu_aDmaBuffer[Channel][0]);",'Correct destionation address for Data

Table continues on the next page...

ID	Subtype	Headline and Description
		in Signal Measurement with DMA mode.
ENGR00373399	NewWork	[ICU] Use "READONLY" instead of "EDITABLE" in xdm file 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: Customer Request Proposed solution (Optional): Use "READONLY" instead of "EDITABLE" in xdm file","File modified: - Icu.xdm
ENGR00364239	NewWork	[ICU] Version checking is missing in configuration template files 'NewWork Description: 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"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJOR"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO

Table continues on the next page...

ID	Subtype	Headline and Description
		R"!][!// [!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!][!// Requirement source: N/A Proposed solution (Optional): Please see Flexray, Adc driver, they are using *_VersionCheck.m. after that include this macro into template files","'File modified: - Icu.mak Adding some files: - Icu_VersionCheck_Inc.m - Icu_VersionCheck_Src.m - Icu_VersionCheck_Src_PB.m
ENGR00369819	Defect	[LIN] ABRQ bit set twice in Lin_Linflex_GoToSleep 'Problem detailed description (how to reproduce it): MPC577XM MCAL4.0 FBR1.1.0 In Lin_Linflex_GoToSleep, the ABRQ is set twice. /* Stop any ongoing transmission */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_BIT_SET32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR2_ABRQ_MASK_U32); /* wait till Cancellation of current frame */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_BIT_SET32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR2_ABRQ_MASK_U32); Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Remove one line with ABRQ set bit. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please remove one line with ABRQ set bit and try to run test.","'Changed file: - Lin_LinFlex.c

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00373452	NewWork	[LIN] Add support for FIBEx import 'NewWork Description: Add a requirement for FR, CAN, ETH and LIN drivers to support FIBEX import Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Added support for FIBEx import
ENGR00369366	NewWork	[LIN] Add traceability comments in the code 'NewWork Description: Add traceability comments in the code Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add traceability comments in Lin.xdm for CPR-MCAL-753.lin",'Lin.xdm
ENGR00376354	Defect	[LIN] Break delimiter shall be two bits to be compatible with Lin2.1 protocol and errata 'Problem detailed description (how to reproduce it): Because the bit LINFlexD_LINCR2[TBDE] is not set in Lin_Linflex_InitChannel In Lin driver, the break delimiter will be one bit time. However according to LIN 2.1 specification and errata: It is strictly recommend using 2 bits for the break delimiter. This is also stated in chapter 59.3.1.3 Break of RM: The break delimiter must be of two-bit duration (to be compliant with LIN protocol 2.1). CE's comment: According to "[LIN005] The Lin module shall conform to the LIN 2.1 Protocol Specification as specified in [16]. This applies to LIN 2.1 Master nodes only. (BSW01576, BSW01504, BSW01577)", the break delimiter bit shall be set by the driver to be compatible with Lin 2.1 protocol specification. Observed behavior: Break delimiter is one bit. Expected behavior: Break delimiter shall be two bits Proposed solution (Optional): Set LINFlexD_LINCR2[TBDE] bit in Lin_Linflex_InitChannel",'-Lin_Linflex.c
ENGR00371587	NewWork	[LIN] Correct code review findings 'NewWork Description: There are several global variables in LIN driver that do not follow SMCAL naming convention: - pcLin_Cfg_Ptr - pLin_Channel_Cfg_Ptr

Table continues on the next page...

ID	Subtype	Headline and Description
		- au8Lin_LinChStatus - au8Lin_LinChFrameStatus - au8Lin_LinChFrameErrorStatus - as8Lin_ChannelHardwareMap - au8TransmitHeaderCommand - u8Lin_LinDrvStatus - au8Lin_LinChHeaderStatus - au8LinSduBuffAddress Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.","'Lin.c Lin_NonASR.h Lin_NonASR.c User manual
ENGR00379411	Defect	[LIN] Correct misra error 'Problem detailed description (how to reproduce it): We have seen some Misra errors in Lin_LINFlex.c file: - Rule 11.1: Line 1558 - Rule 16.7: Line 1230, Line 1369 Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional Lin_TC_0026 (Lin_TS_006) Lin_TC_0027 Trigger: N/A Observed behavior: N/A Expected behavior: No misra errors Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]",'Fixed misra rule.
ENGR00371997	Defect	[LIN] Correct misra errors 'Problem detailed description (how to reproduce it): There are several misra errors in Lin_Cfg.c and Lin_PBcfg.c Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: 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): Please see the Analysis tab","'Lin_Cfg.c Lin_PBcfg.c
ENGR00371998	Defect	[LIN] Correct misra errors and compiler warnings 'Problem detailed description (how to reproduce it): There are several misra errors and compiler warnings in Lin_LINFlex.c 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): NA",'Lin_LINFlex.c
ENGR00369208	Defect	[LIN] Correct misra errors in Lin driver 'Problem detailed description (how to reproduce it): We have seen some Misra errors in Lin.c file, because some Misra comments were incorrectly. Preconditions: Generate some test suites then run Misra report with LINT tool version 9.0j Test Case ID (internal TC that caught the defect) - optional Lin_TS_001, , Lin_TS_002, Lin_TS_003 Expected behavior: No error will be seen in the report",'Lin.c
ENGR00369663	Defect	[LIN] Get_LinStatus may return wrongly an error status. 'Problem detailed description (how to reproduce it): The customer reported following issue in MPC577XM MCAL4.0 FBR1.0.0: In Lin_SendFrame function the error status is cleared at line 1061 (Lin.c): /* Reset error status global flag */ au8Lin_LinChFrameErrorStatus[Channel] = LIN_NO_ERROR; Before aborting the current frame transmission complete at line 731 (Lin_LINFlex.c): /** Stop any ongoing transmission */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_BIT_SET32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR2_ABRQ_MASK_U32); /* wait till Cancellation of current frame */

Table continues on the next page...

ID	Subtype	Headline and Description
		while(u32counter <= LIN_TIMEOUT_LOOPS) { /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ if(LINFLEX_LINCR2_ABRQ_MASK_U32!= (REG_READ32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex)) & LINFLEX_LINCR2_ABRQ_MASK_U32)) { u8TmpReturn = (uint8)E_OK; break; } u32counter++; } If the error occurs before the current frame is aborted, the au8Lin_LinChFrameErrorStatus[Channel] will be set to LIN_XXX_ERROR in error interruption function. After the header just transmitted, the Lin_GetStatus may return an error code instead of busy code. Proposal: Clear error flag after line 731 in Lin_LINFlex.c. au8Lin_LinChFrameErrorStatus[Channel] = LIN_NO_ERROR; Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Lin_GetStatus may return wrongly an error code Expected behavior: Lin_GetStatus returns correct status 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.","'<Description of changes, e.g. Check was added to ... The function xyz was modified because ...> Files Modified: <e.g. Adc_Irq.c > <...>",ENGR00371122
ENGR00378335	NewWork	[LIN] 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.",'- Lin_NonASR.c file.,ENGR00381033
ENGR00368158	NewWork	[LIN] 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. 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): Please see the analysis tab.","'AUTOSAR_MCAL_LIN_UM.pdf um-xml.zip AUTOSAR_MCAL_LIN_UM.xml \topics\revision_history.xml \topics\sub_driver\runtime_errors.xml
ENGR00370001	NewWork	[LIN] Improvement of data type for loop counter variable 'NewWork Description: The customer reported following improvement in MPC577XM MCAL4.0 FBR1.1.0: In Lin_TS_T2D36M10I0R0\src\Lin_LINFlex.c line L909 L927 L940: The u16Loop is uint16, but the loop is uint8. u16loop could be something like u8Loop, isn't it? Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.",'Lin_LINFlex.c
ENGR00375552	Defect	[LIN] Incomplete channel's initialization 'Problem detailed description (how to reproduce it): Lin_Linflex_InitChannel() has not addressed all interrupts and operation modes of LINFLEX. Issues may occur if before Lin_Linflex_InitChannel() is called, LINFLEX is already used for another functionality. Preconditions: Some bits in LINIER, LINCR1 register is not initial default value. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: Initial these bits with default value. 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",'Lin_LINFlex.c
ENGR00368569	Defect	[LIN] 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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];','Lin_LINFlex.c,ENGR00368713
ENGR00365670	Defect	[LIN] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): 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

Table continues on the next page...

ID	Subtype	Headline and Description
		in order to implement all of the following topics that apply : 1. Replace: - ArMajorVersion with ArReleaseMajorVersion - ArMinorVersion with ArReleaseMinorVersion - ArPatchVersion with ArReleaseRevisionVersion 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 Preconditions: N/A 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): Please see above solution.","'Lin_VersionCheck_Inc.m Lin_VersionCheck_Src_PB.m Lin_VersionCheck_Src.m
ENGR00372074	Defect	[LIN] Invalid hardware channel ID in Lin_SetClockMode 'Problem detailed description (how to reproduce it): When we call function Lin_SetClockMode(), in below line: for(u8chLoop=(uint8)0U; u8chLoop < LIN_HW_MAX_MODULES; u8chLoop++) { /* Get the hardware Lin channel from logical channel */ u8Lin_LINFlex = Lin_pChannelConfigPtr[u8chLoop]->u8LinChannelID; ..... The value of u8Lin_LINFlex is id of channel instead of logical channel, so the clock mode will be set to wrong channel. Preconditions: Setup the configuration with channel ID is 0 with logical channel is 1 (Linflex_1) and reverse. RaceRunner Ultra have only one Linflex channel, so we only need configure one channel. Test Case ID (internal TC that caught the defect) - optional tc_api_lin_00101 Expected behavior: Clock mode should be set with the correct logical channel.','Lin_NonASR.c
ENGR00365639	Defect	[LIN] LIN Timeout event cannot be detected 'Problem detailed description (how to reproduce it): Currently, LIN driver on Matterhorn CUT2.0B (with errata e7269) cannot detect

Table continues on the next page...

ID	Subtype	Headline and Description
		timeout events (Slave not response or short response) Preconditions: See above. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: LIN driver cannot detect timeout events Expected behavior: LIN driver can detect timeout events 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",'IPV_LINFlex.c
ENGR00371919	Defect	[LIN] LINESR_BDEF (SDEF), LINESR_IDPEF, LINESR_SFEF will not set in master mode. 'Problem detailed description (how to reproduce it): According to RM, LINESR_BDEF, LINESR_IDPEF, LINESR_SFEF will occur during header reception. These flags will never set in master mode. A header error is an error during the header reception. The error types are: 1. Sync Del error (SDEF) 2. Sync field error (SFEF) 3. Identifier Parity error (IDPEF) However, LIN driver is checking this error status. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional Lin_TC_0023 Trigger: N/A Observed behavior: LIN driver is checking some error status bits which no set in Master mode. Expected behavior: Remove 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",'Lin_LINFlex.c
ENGR00371971	Defect	[LIN] LIN_TX_HEADER_ERROR cannot be detected 'Problem detailed description (how to reproduce it): When a bit error event occurs in header transmission, LIN_TX_ERROR is reported by LIN driver instead of LIN_TX_HEADER_ERROR Preconditions: In Lin_Linflex_InitChannel(), LINCR2.IOBE is set. Consequently, when a bit error event happen, LIN state machine is reset and changed to IDLE mode (LINSR.LINS = 2) -> Cannot distinguish between LIN_TX_HEADER_ERROR and LIN_TX_ERROR

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect): tc_fnc_lin_00208 Trigger: Send a frame with header error and get status of LIN Observed behavior: LIN_TX_HEADER_ERROR Expected behavior: LIN driver should be detected correctly. 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",'LIN_LINFlex.c
ENGR00363001	NewWork	[LIN] Lin driver disable Lin Frame time-out feature. 'NewWork Description: Currently, The Lin driver is using Lin frame time-out feature (in Lin Mode or Output Compare mode) to detect the Lin Short-Response (the frame time is longer than TFrame_Maximum) or Lin No-Response error (No response byte received within TFrame_Maximum). However, according to Lin protocol 2.0/2.1, the Nodes shall not check this time (TFrame_Maximum). This mean the driver should support for the frame which is longer than TFrame_Maximum. Below is apart of Lin protocol specification revision 2.1 November 24, 2006 Page 32 "Tools and tests shall check the TFrame_Maximum. Nodes shall not check this time. The receiving node of the frame shall accept the frame up to the next frame slot (i.e. next break field), even if it is longer then TFrame_Maximum" Can Lin driver support the frame which is longer than TFrame_Maximum? The idea is that adding a new parameter (e.g Frame Timeout Disable) at the configuration time. 1) If Frame Timeout Disable is OFF (by default), the Lin driver will behave as current implementation (The Lin frame must be within TFrame_Maximum). 2) If Frame Timeout Disable is ON, the Lin driver does not enable frame timeout feature, the master will accept the frame that is longer than TFrame_Maximum The limitation: When Frame Timeout Disable is ON, the Short Response and No Response Error will not be supported by Lin driver. Requirement source: CPR-MCAL-753.lin (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.","'Make Frame Timeout feature become optional, in details: + Update Lin.xdm: add LinDisableFrameTimeout entry + Update Lin_Cfg.h: add macro LIN_DISABLE_FRAME_TIMEOUT + UM, IM + Add testsuit Lin_TS_018
ENGR00363099	NewWork	[LIN] Lin driver disable Lin Frame time-out feature.

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: Adding a new parameter (e.g Frame Timeout Disable) at the configuration time. 1) If Frame Timeout Disable is OFF (by default), the Lin driver will behave as current implementation (The Lin frame must be within TFrame_Maximum). 2) If Frame Timeout Disable is ON, the Lin driver does not enable frame timeout feature, the master will accept the frame that is longer than TFrame_Maximum The limitation: When Frame Timeout Disable is ON, the Short Response and No Response Error will not be supported by Lin driver. Requirement source: CPR-MCAL-753.lin (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.",'Disable Output Compare Interrupt and Idle on timeout feature of LINFlexD.
ENGR00379167	Defect	[LIN] Lin_Linflex_au8StateOfBEF should be initialized when Lin_Init is called. 'Problem detailed description (how to reproduce it): The Lin_Linflex_au8StateOfBEF is not initialized when Lin_Init is called. If this is not initialized, after Module Init, if the function Lin_GetStatus is called, the data from Lin_Linflex_au8StateOfBEF can be used. So it relays on the initialization values from the compiler in case no updates were done in the meantime. Please initialize the this variable when Lin_Init is called. Preconditions: Call function Lin_Init() Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Lin_Linflex_au8StateOfBEF is not initialized. Expected behavior: Lin_Linflex_au8StateOfBEF is initialized. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add line: /* Clear State of Bit Error flag */ Lin_Linflex_au8StateOfBEF[u8Channel] = 0u; to function Lin_Linflex_InitChannel().",'- Lin_Linflex.c
ENGR00366839	Defect	[LIN] Missing MemMap declarations in driver files 'Problem detailed description (how to reproduce it): The following variables do not have consistent memory section guarding for definition and all other extern declarations: Eth_InternalCfgPtr Fls_ACEraseRomEnd Fr_FrDemCtrlTestResultPtr Lin_Config_PC Mcu_Config_PC

Table continues on the next page...

ID	Subtype	Headline and Description
		Port_pConfig Wdg_au8Index Wdg_pConfigPC the variables shoudl be allocted in the same MemMap.h section where it is defined, and in every other place where it's declared external. this causes a link error when sda=all compiler option is used. Preconditions: No special precondition. Compiler/linker problem. Test Case ID (internal TC that caught the defect) - optional NA Compiling the code Observed behavior: Linker error. Expected behavior: No errors. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add Lin_Config_PC in the right memory map section",'Lin.c
ENGR00372699	Defect	[LIN] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. 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): Please see the Analysis tab","'Lin_VersionCheck_Inc.m Lin_VersionCheck_Src.m Lin_VersionCheck_Src_PB.m Lin.mak
ENGR00369818	Defect	[LIN] Response transmission shall be stopped after bit error 'Problem detailed description (how to reproduce it): The customer reported following defect in LIN driver. In Lin_Linflex_InitChannel function the bit IOBE is cleared and not set anywhere later. If the bit error occurs in response, the frame transmission will be continue. However the LIN240 stated that the transmision should be aborted. [LIN240] In case of response transmission errors, the LIN 2.1 specification describes within the frame processor state machine how to handle such errors. It is stated that a mismatch between sent and readback data shall be detected not later than after the completion of the byte field containing the mismatch.

Table continues on the next page...

ID	Subtype	Headline and Description
		Furthermore, LIN 2.1 specifies that the transmission shall be aborted. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional tc_fnc_lin_00214 Trigger: N/A Observed behavior: N/A Expected behavior: If the bit error occurs in response, the frame transmission will be continue. 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",'Lin_LINFlex.c
ENGR00376383	Defect	[LIN] Sleep command may not be sent successfully 'Problem detailed description (how to reproduce it): In Lin_GoToSleep function, if there is a frame transmission complete interrupt between setting au8TransmitHeaderCommand[channel] to LIN_TX_SLEEP_COMMAND and setting aborting the current frame, the channel is wrongly set into LIN_CH_SLEEP_STATE and the SLEEP bit is also set LINFLEX_LINCR1. Consequently, the sleep command will not be sent in Lin_lpw_GoToSleep. The au8TransmitHeaderCommand[channel] shall set to LIN_TX_SLEEP_COMMAND after aborting current channel sucessfully and before setting LINFLEX_LINCR2_HTRQ. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional Lin_TC_0026 (Lin_TS_006) Lin_TC_0027 Trigger: N/A Observed behavior: N/A Expected behavior: No misra errors 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.",'- Lin.c,ENGR00376831
ENGR00373369	NewWork	[LIN] Some Non-Autosar configuration parameters should have OPTIONAL attribute 'Summary the discussion between TL and CE: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Implementation details The parameters that can be excluded from a specific configuration shall be marked as optional (<a:a name="OPTIONAL" value="true"/>, which results in lower multiplicity =0 and upper multiplicity 1). The AUTOSAR parameters have already defined the lower and the upper multiplicity. This change is needed only for the NON-Autosar parameters. The boolean Non-Autosar parameters that enable / disable Non-Autosar features must not be marked as OPTIONAL - only the additional configuration parameters that are needed only when Non-Autosar features are enabled. All parameters that are under this category shall be marked as OPTIONAL=true in the xdm. The check that the features were correctly configured shall be transferred to the boolean parameter that enables the feature. Example: Node1 - configuration node for Non-Autosar feature that selects between transfer mode interrupt or DMA should not be OPTIONAL Node2 - configuration node that selects Dma channel reference needed when DMA transfer is selected should be OPTIONAL when Node1 selects DMA, there should be added an invalid check for the existence of Node2 Reference point SpiPhyTxDmaChannel will be enabled when SpiPhyUnitAsyncMethod is set up to be DMA the changes: into <v:var name="SpiPhyUnitAsyncMethod" type="ENUMERATION"> [...to be added the check that everything is configured the INVALID tag] <a:da name="INVALID" type="XPath"> <a:tst expr="((node:value(.)='DMA') and (node:exists(../SpiPhyTxDmaChannel))) or not(node:value(.)='DMA')" false="Please enable SpiPhyTxDmaChannel"/> </a:da> <v:ref name="SpiPhyTxDmaChannel" type="CHOICE-REFERENCE"> [...] <a:a name="OPTIONAL" value="true"/> <a:da name="EDITABLE" type="XPath" expr="not(node:value(../SpiPhyUnitAsyncMethod) = 'DMA')"/> ---------- Original request from customer engineer---------- Parameters: - Non-autosar parameters. - Specific parameters under certain conditions. This is my additional details: Customer don't use some reference nodes (DMA with Spi, Clock reference with GPT...). Epc file that exported from EB tresos with no reference value in DMA SPI case. For ex: <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyRxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannel</DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE>

Table continues on the next page...

ID	Subtype	Headline and Description
		<DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannelAux</DEFINITION-REF> </ECUC-REFERENCE-VALUE> But in EPD file, these parameters are defined as mandatory with lower/upper multiplicity =1. So they should be have default value because a mandatory parameter cannot be empty. Proposed solution: 1. The not used parameters _should_not_ be part of the Tresos Export. So please change the xdm-file in that way that the parameters will not be part of the export for the case that they are not used! 2. Lower multiplicity =0 Upper multiplicity =1 for optional. ----------",'Lin.xdm
ENGR00375066	Defect	 [LIN] Some bits are only programmed when the slave mode is supported 'Problem detailed description (how to reproduce it): The customer reported following issue in LIN drivers FBR1.1.0 Matterhorn MCAL4.0 The RM's note: The bit is not programmed when generic slave = 0 or no_of_filters = 0 In the function Lin_Linflex_InitChannel, the BF bit is set at line 480. REG_BIT_SET32(LINFLEX_LINCR1_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR1_BF_MASK_U32); And it is cleared at line 484. REG_BIT_CLEAR32(LINFLEX_LINCR1_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR1_BF_MASK_U32); There is also the same issue with the bits: LINFlexD_LINCR1 MME LINFlexD_LINIER HEIE LINFlexD_LINSR HRF LINFlexD_LINSR SFEF LINFlexD_LINSR IDPEF CE's comment: Those bits are only programmed when slave mode is supported by the LinFlex module. Maybe there would be no function problem since the bits are readonly. The driver should set or clear those bits only when the slave mode is supported. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional Lin_TC_0023 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): Please see the analysis tab.",'Lin_LINFlex.c

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00366974	Defect	[LIN] Some variables are not referenced when LIN_DISABLE_FRAME_TIMEOUT is ON 'Problem detailed description (how to reproduce it): There are several compiler warnings in Lin_LINFlex.c. When LIN_DISABLE_FRAME_TIMEOUT is defined as STD_ON, several variables (u32LINOCR_OC in Lin_Linflex_SendHeader, Lin_Linflex_SendResponse, Lin_Linflex_GoToSleep, Lin_Linflex_ErrorInterruptHandler, and u32CompareResult in Lin_Linflex_ErrorInterruptHandler) will not be referenced. Preconditions: See above. 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): Add the guard (LIN_DISABLE_FRAME_TIMEOUT == STD_OFF) to their declaration","'When LIN_DISABLE_FRAME_TIMEOUT is defined as STD_ON, several variables (u32LINOCR_OC in Lin_Linflex_SendHeader, Lin_Linflex_SendResponse, Lin_Linflex_GoToSleep, Lin_Linflex_ErrorInterruptHandler) will not be referenced. Add the guard (LIN_DISABLE_FRAME_TIMEOUT == STD_OFF) to their declaration
ENGR00372312	Defect	[LIN] The Lin frame status and errors status should be updated right after frame abortion and before new header request 'Problem detailed description (how to reproduce it): The customer reported following issues in FBR1.1.0 HF2. The HF2 updates frame status and errors status after frame abortion by ENGR00369663 and ENGR00369627. However in HF2, the frames status and errors status are updated after new header transmision request. This may causes a wrongly updating status if error occurs before frames status&error status are updated. The Lin frame status and errors status should be updated right after frame abortion sucessfull and before new header request. e.g: ..... /** Stop any ongoing transmission */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_BIT_SET32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex), LINFLEX_LINCR2_ABRQ_MASK_U32); /* wait till Cancellation of current frame */ while(u32counter <= LIN_TIMEOUT_LOOPS) { /**

Table continues on the next page...

ID	Subtype	Headline and Description
		* @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ if(LINFLEX_LINCR2_ABRQ_MASK_U32!= (REG_READ32(LINFLEX_LINCR2_ADDR32(u8Lin_LINFlex)) & LINFLEX_LINCR2_ABRQ_MASK_U32)) { u8TmpReturn = (uint8)E_OK; break; } u32counter++; } if((uint8)E_OK == u8TmpReturn) { /* Reset error status global flag */ au8Lin_LinChFrameErrorStatus[Channel] = LIN_NO_ERROR; /* Set header status to TRUE if master wants to send the response frame */ if (LIN_MASTER_RESPONSE == PduInfoPtr->Drc) { au8Lin_LinChHeaderStatus[Channel]= (uint8)TRUE; au8TransmitHeaderCommand[Channel] = LIN_TX_MASTER_RES_COMMAND; /* Update LIN channel frame operation status to LIN_CH_READY_STATE */ au8Lin_LinChFrameStatus[Channel] = LIN_CH_READY_STATE; } else { au8TransmitHeaderCommand[Channel] = LIN_TX_SLAVE_RES_COMMAND; au8Lin_LinChFrameStatus[Channel] = LIN_CH_RECEIVE_NOTHING_STATE; } /* Release the Message buffer */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_WRITE32(LINFLEX_LINSR_ADDR32(u8Lin_LINFlex), LINFLEX_LINSR_RMB_MASK_U32); /* Clear the Noise Flag of LIN Error Status Register */ /** * @violates @ref Lin_LINFlex_c_REF_3 cast from unsigned int to pointer */ REG_WRITE32(LINFLEX_LINESR_ADDR32(u8Lin_LINFlex), LINFLEX_LINESR_NF_MASK_U32); ..... Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The Lin frame status and errors status should be updated right after frame abortion and before new header request 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
		Please see the analysis tab.",'Lin.c,ENGR00372500
ENGR00374998	Defect	[LIN] The LinChannelWakeUpSupport should be checked before checking WUF flag and calling the EcuM_SetWakeupEvent 'Problem detailed description (how to reproduce it): The customer reported following issue in Lin driver of MPC5777M FBR1.1.0. In function Lin_Linflex_InitChannel: if(LINFLEX_LINSR_WUF_MASK_U32== (REG_READ32(LINFLEX_LINSR_ADDR32(u8Lin_LINFlex)) & LINFLEX_LINSR_WUF_MASK_U32)) { /* The wake-up shall directly be reported to the EcuM */ /* via EcuM_SetWakeupEvent call-back function. */ EcuM_SetWakeupEvent(pLin_Channel_Cfg_Ptr[u8Channel]- >LinChannelEcuMWakeupSource); } The EcuM_SetWakeupEvent is called without checking LinChannelWakeUpSupport According to LIN250, the parameter LinChannelWakeUpSupport should be checked [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.() Symbolic names of the available configuration sets are provided by the configuration description of the LIN driver. See chapter 10 about configuration description. Preconditions: Generate plugin and build the test Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: The driver should be compliance with LIN250 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In Lin_Linflex_InitChannel(), in the case that WUF is set, checking LinChannelWakeUpSupport before calling EcuM_SetWakeupEvent.",'Lin_LINFlex.c
ENGR00375092	Defect	[LIN] The bit LINCR1_SLEEP must be cleared before clearing UARTCR_UART 'Problem detailed description (how to reproduce it): The customer reported following issue in Lin driver FBR1.1.0 MPC5777M MCAL4.0 In in_LINFlex.c, the line 450 clears the UARTCR_UART bit but after setting INIT bit from reset, the SLEEP bit is 1. The UARTCR_UART is not cleared because according to RM, the UARTCR_UART is written only in initialization

Table continues on the next page...

ID	Subtype	Headline and Description
		mode. Preconditions: Call Lin_Init() and check register status. Test Case ID (internal TC that caught the defect) - optional Lin_TC_0023 Expected behavior: The bit LINCR1_SLEEP must be cleared Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In Lin_Linflex_InitChannel(), LINCR1.SLEEP must be cleared before clearing UARTCR.UART.','Lin_LINFlex.c
ENGR00369627	Defect	[LIN] The channel status may be wrongly updated by Lin interrupt 'Problem detailed description (how to reproduce it): Please see the below report from our customer on MPC577XM MCAL4.0 FBR1.1.0 When the timeout feature is off: In the Lin_SendFrame, the au8Lin_LinChFrameStatus[] is set to LIN_CH_READY_STATE or LIN_CH_RECEIVE_NOTHING_STATE depending on frames before aborting current frame However if the current transmission just completed/or error occur after au8Lin_LinChFrameStatus[] set, it will be updated again in the interrupt (RxTx interrupt or error interrupt). This leads to the problem that the frame status is not correct. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Lin_GetStatus returns wrong status. Expected behavior: Lin_GetStatus returns correct status. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update au8Lin_LinChFrameStatus[] after aborting the current frame successfully.","'Changed file: - Lin.c
ENGR00370118	NewWork	[LIN] Update Base address array to display all HW Units 'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif

Table continues on the next page...

ID	Subtype	Headline and Description
		#ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the adress 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....",'Updated the Base Address array.,ENGR00371442
ENGR00369263	Defect	[LIN] Update driver information in user manual 'Problem detailed description (how to reproduce it): Currently, the user manual of LIN driver is missing a lot of information about Defines, Enums, Structs, Types and Variables. Preconditions: UM missing information 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 driver information in user manual","'AUTOSAR_MCAL_LIN_UM.pdf docgen\um-xml.zip um-xml\AUTOSAR_MCAL_LIN_UM.xml um-xml\topics\sub_driver\sub_doxygen\

Table continues on the next page...

ID	Subtype	Headline and Description
		um-xml\topics\sub_driver\sub_doxygen\sub_variables\ um-xml\topics\sub_driver\sub_doxygen\sub_structs\ um-xml\topics\sub_driver\sub_doxygen\sub_enums\ um-xml\topics\sub_driver\sub_doxygen\sub_defines\
ENGR00373400	NewWork	[LIN] Use "READONLY" instead of "EDITABLE" in xdm file 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.",'update EDITABLE tag with READONLY
ENGR00378354	NewWork	[LIN] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see the analysis tab.","'Will be note when implement this CR: - Lin.c

Table continues on the next page...

ID	Subtype	Headline and Description
		- Lin.h - Lin_Cfg.c - Lin_PBcfg.c - Lin_IPW.c - Lin_IPW.h",ENGR00381218
ENGR00376062	NewWork	[MCL] Add Mcl_DeInit function to the MCL driver 'NewWork Description: Add Mcl_DeInit function to the MCL driver. The requirement for the new API: "The API Mcl_DeInit shall deinitialize the underlaying hardware" Requirement source: Requested by client. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional):",'Files modified:,ENGR00378463 ENGR00378465
ENGR00367784	NewWork	[MCL] Add support for the crossbar control (AXBS IP) 'NewWork Description: Add support for the crossbar control (AXBS IP). As per ITWG meeting 08.01.2016 only static configuration is needed(by MclInit).No runtime functions are needed.They will be implemented only by customer request. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add support for the crossbar control, add configuration in Mcl_Init.",'Add support for the crossbar control (AXBS IP)
ENGR00373849	NewWork	[MCL] Add the crossbar switch to the MCL design 'NewWork Description: Add the crossbar switch to the MCL design (MCl_Init function, plugin, IPV_AXBS). Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Add the crossbar switch to the MCL design
ENGR00375847	Defect	[MCL] Code size not optimized in Mcl_Dma_Irq.c 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		In file Mcl_Dma_Irq.c, function Mcl_DmaProcessCombinedInterrupt still is built although in some platforms it doesn't use. So it make some issue relate to LDRA Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: In file Mcl_Dma_Irq.c, function Mcl_DmaProcessCombinedInterrupt still is built although in some platforms it doesn't use. So it make some issue relate to LDRA Expected behavior: DmaProcessCombinedInterrupt won't be built in some platform that doesn't use 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 Macro "#if (MCL_DMA_COMBINED_TR_COMPLETION_ISR == STD_ON)" to guard it","'File Modified : - Mcl_Dma_Irq.c
ENGR00373121	NewWork	[MCL] Comment compiler warnings 'NewWork Description: Comment compiler warnings.",'Comment compiler warnings
ENGR00367278	Defect	[MCL] Correct compiler warnings 'Problem detailed description (how to reproduce it): With some compiler options (S32K EAR release compiler options) we get the following compiler warnings: warning #47-D: incompatible redefinition of macro "DMA_DCHPRI_ECP" warning #47-D: incompatible redefinition of macro "DMA_DCHPRI_DPA" Preconditions: S32K EAR release compiler options Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Compiler warnings observed. Expected behavior: The target is not to have 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): Change DCHPRI defines from uint32 to uint8: Initial: #define DMA_DCHPRI_ECP (0x00000080UL) #define DMA_DCHPRI_DPA (0x00000040UL) After update: #define DMA_DCHPRI_ECP ((uint8) 0x80UL) #define DMA_DCHPRI_DPA ((uint8) 0x40UL)","'File modified:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Reg_eSys_Dma.h
ENGR00363038	Defect	[MCL] DMA_UPDATE_MAJOR_LINKCH will configure incorrect linked channel 'Problem detailed description (how to reproduce it): For DMA channel ID higher than 31, the DMA_UPDATE_MAJOR_LINKCH will configure incorrect linked channel. The defect affects only platforms with more than 31 channels per DMA instance available. Preconditions: Hardware channel ID higher than 31. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: For DMA channel ID higher than 31, the DMA_UPDATE_MAJOR_LINKCH will configure incorrect linked channel; instead of configuring the actual channel ID, it will always confiure it as 31. The reason is the mask MAJOR_LINKCH_MASK_U32 is not correctly defined for platforms with more than 31 channels on DMA instance. Expected behavior: Configure the channel ID given as input. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update Mcl_DMa_Types to define the correct mask for the define MAJOR_LINKCH_MASK_U32: #if (MCL_DMA_NB_CHANNELS == 0x40U) #define MAJOR_LINKCH_MASK_U32 ((uint32)(0x3FUL)) #else #define MAJOR_LINKCH_MASK_U32 ((uint32)(0x1FUL)) #endif","'Files modified : - Mcl_Dma_Types.h
ENGR00377643	Defect	[MCL] Driver limitations are not described in the user manual 'Problem detailed description (how to reproduce it): The driver limitations are missing from user manual. The following DMA hardware features are partially implemented or not implemented by MCL: ? eDMA_CR is partially implemented: THE EMLM bit can not be configured; the EMLM bit is always set to 1 during the DMA initialization. ? eDMA_HRS is not implemented: MCL does not provide API to retrieve this information. On GTM platforms the following information is missing: The following GTM hardware features are partially implemented or not implemented by MCL: ? GTMINT_GTMMCR is partially implemented: the AISREN bit can not be configured; the AISREN bit is always written as 0 during the GTM initialization. ? GTM_CTRL is partially implemented: the TO_VAL, TO_MODE, RF_PROT bits can not be configured; the TO_VAL, TO_MODE, RF_PROT bits are always

Table continues on the next page...

ID	Subtype	Headline and Description
		written as 0 during the GTM initialization. ? GTM_BRIDGE_MODE is partially implemented: the BRG_RST, BUFF_OVL, MSK_WR_RSP, BRG_MODE bits can not be configured; BRG_RST, BUFF_OVL, Driver Limitations User Manual, Rev. 1.3 88 NXP Semiconductors MSK_WR_RSP are always written as 0 during the GTM initialization. BRG_MODE is always written as 1 during the GTM initialization. ? TBU_CH0_BASE is partially implemented: Register is initialized and deinitialized. MCL does not provide API to retrieve this information. ? TBU_CH1_BASE is partially implemented: Register is initialized and deinitialized. MCL does not provide API to retrieve this information. ? TBU_CH2_BASE is partially implemented: Register is initialized and deinitialized. MCL does not provide API to retrieve this information. ? GTMINT_GTMINTCLR is not implemented. ? GTMINT_GTMAEICR is not implemented. ? All other GTM registers(except the ones above and GTM_RST, CMU_CLK_EN, CMU_GCLK_NUM, CMU_GCLK_DEN, CMU_FXCLK_CTRL, CMU_CLK_n_CTRL, TBU_CHEN, TBU_CH0_CTRL, TBU_CH1_CTRL, TBU_CH2_CTRL) are not implemented. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Driver limitations are not described in the user manual Expected behavior: Driver limitations should be described in the user manual Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update user manual with known driver limitations.','Write the limitations in the user manual
ENGR00371879	Defect	 [MCL] ELINK and LINKCH should be masked out in the Mcl_DmaTcdGetIterCount 'Problem detailed description (how to reproduce it): Mcl_DmaTcdGetIterCount returns incorrect CITER value in case CITER_ELINK is 1. Preconditions: CITER_ELINK set to 1 in TCD word 6 Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: In the Mcl_DmaTcdGetIterCount, all 16 bit (eDMA_TCDn_CITER_ELINKYES/ ELINKNO) is returned. While CITER Field: ELINK YES 7-15 Bit ELINK NO 1-15 Bit In order to not get the ELINK and LINKCH, these field should be masked out. The driver should check ELINK bit then mask LINKCH from Citer value if ELINK==1. Expected behavior: Mcl_DmaTcdGetIterCount returns correct CITER value in all cases. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): When reading TCD word 6, if CITER_ELINK is 1, then mask the CITER_ELINK and CITER_LINKCH bits.','ELINK bit is not checked when getting Citer value.,ENGR00372320
ENGR00378340	Defect	[MCL] Exclusive areas are missing 'Problem detailed description (how to reproduce it): Missing exclusive area to protect the following global data: TCDx.2ND_WORD 32 bits (SMOD SSIZE DMOD DSIZE SOFF) TCDx.6TH_WORD 32 bits (CITER.LINKCH CITER.E_LINK CITER DOFF) TCDx.8TH_WORD 32 bits (BITER.LINKCH BITER.E_LINK BITER BWC MAJOR LINKCH FLAGS) DMA_CR DMA_DCHPRIn Preconditions: In case Dma_SetChannelPriority may interrupt Dma_Init or Dma_DeInit In case Dma_Init or Dma_DeInit may interrupt Dma_SetChannelPriority In case APIs which configure the TCDs words 2,6,8 may interrupt each other. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Exclusive areas are missing. This might lead to inconsistencies. Expected behavior: Exclusive area must be used to protect critical sections. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Exclusive area must be used to protect critical sections.','Implement data consistency mechanism (exclusive areas).,ENGR00379083
ENGR00368168	NewWork	[MCL] Improve Dem/Det error description in Usermanual 'NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that

Table continues on the next page...

ID	Subtype	Headline and Description
		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.",'Improved description of DEM/DET error codes description in IM/UM
ENGR00380925	NewWork	[MCL] Improve the APIs which set and get CITER and BITER 'NewWork Description: Cuurently the APIs have un-uniform approach(16 bit or 32 bit access, rmw or reg_Write). Improve the code so that the functions -use similar approach -don't directly access underlaying hardware, call macros from IPV_DMA to do that (use directly reg_Read, rmw or reg_Write) -in IPV_DMA: access only on 32 bits and use rmw or access only on 16 bits and use the doubled ralative macros for little and big endian Update exclusive areas if needed. To update the IPW access for APIs: Mcl_DmaTcdSetIterCount Mcl_DmaTcdSetLinkAndIterCount Mcl_DmaTcdGetIterCount Mcl_DmaUpdateIterCount Mcl_DmaGetCrtIterCount Mcl_DmaGetStartIterCount Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","'Files Modified: - Mcl_IPW.c - Mcl_Dma.c
ENGR00379261	NewWork	[MCL] Improve the TCD macro access 'NewWork Description: For accesing the TCD words on 32 bytes, MCL uses 2 sets of offset define which have the same value and the same meaning (word offset). Please replace the logical offsets with the word offests and delete the logical offsets: Example: DMA_TCD_TA_SOFF_OFFSET_U32 - >DMA_TCD_2ND_WORD_OFFSET_U32. DMA_TCD_DLAST_SGA_OFFSET_U32 - >DMA_TCD_7TH_WORD_OFFSET_U32 DMA_TCD_DADDR_OFFSET_U32 ->DMA_TCD_5TH_WORD_OFFSET_U32 DMA_TCD_DOFF_OFFSET_U32-> DMA_TCD_6TH_WORD_OFFSET_U32 DMA_TCD_CITER_WORD_OFFSET_U32-> DMA_TCD_6TH_WORD_OFFSET_U32 DMA_TCD_TA_SOFF_OFFSET_U32->

Table continues on the next page...

ID	Subtype	Headline and Description
		DMA_TCD_2ND_WORD_OFFSET_U32 DMA_TCD_MLNO_OFFSET_U32-> DMA_TCD_3RD_WORD_OFFSET_U32 DMA_TCD_SLAST_OFFSET_U32 -> DMA_TCD_4TH_WORD_OFFSET_U32 Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please replace the logical offsets with the word offests and delete the logical offsets.","'Files Modified : - Reg_eSys_Dma.h - Mcl_Dma.c
ENGR00379348	NewWork	[MCL] Improvement the DeInit function 'NewWork Description: De-initialize at least CITER, BITER, INT_MAJ and INT_HALF. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","'Files modified: Mcl_Dma.c
ENGR00373419	Defect	[MCL] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in our MCAL drivers. Variable declarations should have the definitions and all extern declarations mapped to the same MemMap sections. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: DMA function notification declarations is not mapped to MCL_START_SEC_CODE section. Expected behavior: All global variables used by Mcal drivers must be mapped to the special 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): Guard all variable definitions and declarations with the appropriate MemMap section. DMA function notification declarations should be mapped to MCL_START_SEC_CODE section.','Inconsistent/Incorrect usage of Memory Allocation Sections

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00372409	NewWork	[MCL] Internal Refactoring: move MclErrorNotificationDma to the config structure 'NewWork Description: Currently the MclErrorNotificationDma is generated as define. Move it the the configuration structure. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Internal Refactoring: move MclErrorNotificationDma to the config structure.",'Internal Refactoring: move MclErrorNotificationDma to the config structure,ENGR00372459
ENGR00371548	NewWork	[MCL] Internal factoring:rework the configurations for DMA - code ticket 'NewWork Description: Background: At the beginning the MCL driver only included DMA, now it includes other features on some platforms -trgmux config -axbs config -gtm config Rework the config structure of MCL(Mcl_ConfigType), group all Dma configs in one structure and rename them to add Dma inside the naming. Now Mcl_ConfigType includes the following DMA configurations: NumChannels pMclChannelsConfig pMclDmaHwIpsConfig HwToLogicChannelMap Move them to a new structure(MclDmaConfig) and rename them: NumChannels->MclDmaNumchannels pMclChannelsConfig->pMclDmaChannelsConfig pMclDmaHwIpsConfig->this is ok HwToLogicChannelMap->MclDmaHwToLogicChannelMap Also consult the code checklist to see that the new namings of the types and structures are correct. Update the configurations according to the changes. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'rework the configurations for DMA
ENGR00375756	NewWork	[MCL] Internal refactoring: improve the interrupt handlers 'NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Improve the interrupt handlers to support all platforms, reduce the number of defines used and make sure that for a specific platform only the needed handlers and ISRs are compiled. Consider each specific platform case: -some platforms have ISR per chanel -other platforms have ISR per instance -cobra has some other approach Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Modify Mcl_Dma_Irq.c to support all platforms, reduce the number of defines used and make sure that for a specific platform only the needed handlers and ISRs are compiled.","'Improve the interrupt handlers to support all platforms, reduce the number of defines used and make sure that for a specific platform only the needed handlers and ISRs are compiled.",ENGR00379964
ENGR00368239	NewWork	[MCL] Internal refactoring: make Mcl more generic for DMA part 'NewWork Description: Synchronze MCL implementation between platforms, move files from specific to generic branches. 1.Make Cdd_Mcl_Cfg.c, Cdd_Mcl_PBCfg.c, and Cdd_Mcl_PBCfg_42.c sto use a similar approach when generating code(same paths). 2.Remove the implementation for DMA with 1 instance, we will keep DmaInstance as list having container having all dma instance parameters for all platforms. 3.Keep MclConfigSet as list even for ASR 4.2 code. 4.Keep the MclIsrAvailable for 4.0.3 and 4.2.1 to have the same implementation between ASR versions. 5.Keep a maximum of all DMAMUX sources (10 dmamux sources) in the xdm, but if a platform has only 4 dmamuxes, for example, the other 6 will always be not editable for user 7.Keep a maximum of group priorities(they will be non editable for platforms which don't use them)",'Internal refactoring: make Mcl more generic for DMA part
ENGR00374365	NewWork	[MCL] LMEM control has to be under MCL responsability 'NewWork Description: LMEM control has to be under MCL responsibility. Thisis because the LMEM integration at SoC level deiffres from SoC to SoC and is very difficult to maintain a generic code in BASE. Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add support for LMEM IP in MCL driver. Create APIs for flush, invalidate, clear, enable and disable. Update all function calls with the new API names throughout the MCL

Table continues on the next page...

ID	Subtype	Headline and Description
		driver.","'Renamed cache_flush_all(), cache_invalidate_all() to Mcl_IPW_CacheInvalidateAll(), Mcl_IPW_CacheFlushAll()
ENGR00379359	Defect	[MCL] MISRA error reported for MCL users when Mcl_DmaTcdSetFlags function is used 'Problem detailed description (how to reproduce it): Prototype for Mcl_DmaTcdSetFlags is declared in CDD_Mcl.h as below: FUNC(void, MCL_CODE) Mcl_DmaTcdSetFlags (P2VAR(Mcl_DmaTcdType, AUTOMATIC, MCL_APPL_DATA) pTcdAddress, VAR(uint8, AUTOMATIC) u8Flags); When the other module use this function like: Mcl_DmaTcdSetFlags(Adc_Adcdigv2_DmaTcdConfigMem[HwIndex][ChIdx], (DMA_TCD_START_U32 | DMA_TCD_E_SG_U32)); because DMA_TCD_START_U32 and DMA_TCD_E_SG_U32 is declared as uint32, so the argument will be converted from uint32 to uint8. Misra 10.1 will be raised. Preconditions: Mcl_DmaTcdSetFlags function is used. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Misra 10.1 will be raised in MCL users. Expected behavior: No MISRA error will be raised in MCL users becasue of MCL. 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",'MISRA error reported in MCL users when Mcl_DmaTcdSetFlags function is used.
ENGR00378250	NewWork	[MCL] Make crossbar implementation generic 'NewWork Description: Make the crossbar implementation generic for the smcal platforms. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Make crossbar implementation generic
ENGR00374483	Defect	[MCL] Mcl_IPW_DmaClearDone does not clear the Done bit 'Problem detailed description (how to reproduce it): Mcl_IPW_DmaClearDone does not clear the Done bit. Root cause: In function Mcl_IPW_DmaClearDone, the usage of macro DMA_CDNE_ADDR32 is wrong: parameter should be DMA instance, not DMA

Table continues on the next page...

ID	Subtype	Headline and Description
		channel. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Mcl_IPW_DmaClearDone does not clear the Done bit Expected behavior: Mcl_IPW_DmaClearDone will clear the Done bit Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA","'File modified: + Mcl_IPW.c
ENGR00369020	NewWork	[MCL] Remove EcuM from IM and tests 'NewWork Description: MCL should not have a dependency on Ecum. Remove EcuM from IM and tests",'Remove EcuM from IM and tests
ENGR00362268	NewWork	[MCL] Review bit field operation 'NewWork Description: the attached MCAL_REG_ACCES_MACRO_LIST.xlsx contains some findings on Matterhorn MCAL 0.9.0 Beta. Please review all usage of REG_BIT_SET and REG_BIT_CLEAR using multi-bit masks for multi-bit bitfields. There are issues with setting the new value if the bitfield in register is not completely zero when using bitwise "OR" like in REG_BIT_SET. Also make sure multi-bit masks used for these macros cover the number of bits in the target bitfields. Customer feedback: For the bit field access (clear/set), it is not recommended to use REG_BIT_XXXX operation. - Check the bit field is cleared before setting. - Check the clear operation is used with correct mask (all bit field is 1). e.g: MPC577XM MCAL0.9.0. Adc driver has use REG_BIT_SET32 to configure trigger edge for hardware trigger conversion. REG_BIT_SET32(ADC_MCR_REG(unit), mcr_set_val); Configure G0 with HW trigger Falling edge. Configure G1 with HW trigger Rising edge. - Enable HW trigger G0 - Disable HW trigger G0 - Enable HW trigger G1 Now the HW trigger G1 will be configured with both edge. Another defect number-ENGR00341737 was also reported by the customer. The defect relates to clear/setting ODC bit field in MCSR registers. Customer engineer recommendation: Don't use REG_BIT_XXXX macro to configure (clear/set) bit field. It is better to use REG_RMWXXX macro with bit field mask to clear/set the bit field. Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		Please review all usage of REG_BIT_SET and REG_BIT_CLEAR using multi-bit masks for multi-bit bitfields. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please review all usage of REG_BIT_SET and REG_BIT_CLEAR using multi-bit masks for multi-bit bitfields. There are issues with setting the new value if the bitfield in register is not completely zero when using bitwise "OR" like in REG_BIT_SET. Also make sure multi-bit masks used for these macros cover the number of bits in the target bitfields. If there are cases where the bitfield might not be zero before an OR operation, use REG_BIT_RMW - but please consider the extra execution time implied by this macro.","Performed review on multi-bit bitfields operations to avoid inconsistencies after successive bit mask setting.
ENGR00373224	Defect	[MCL] Some files rely on indirect inclusion 'NewWork Description: 1.If a file uses something from another header, then it must include that header. Eg:Reg_ESysDmaMux.h must include Cdd_Mcl_Cfg.h becasue of the define MCL_DMAMUX_CHCFG_LITTLE_ENDIAN.Check all files. 2.In Mcl_DmaMux.c the inclusion "#include "Mcl_DmaMux.h" must be guarded by Enable_DMA. (Please make sure Cdd_MCl_Cfg.h is included before using ENABLE_DMA define).Check that all DMA files are guarded by this include and that all Crossbar files are guarded by ENABLE_CROSSBAR. If another include is added, then also make sure to update the file version check also. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]","N/A
ENGR00375898	Defect	[MCL] Update function Mcl_DmaGetCrtIterCount 'Problem detailed description (how to reproduce it): Currently, the function Mcl_DmaGetCrtIterCount get all 16 bit in Current major iteration count. Actually, it need get only citer bits (do not get CITER.E_LINK). There should be similar to Mcl_DmaGetStartIterCount (For more details, refer Mcl_IPW_DmaGetChBiter) Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		The function Mcl_DmaGetCrtIterCount get all 16 bit in Current major iteration count. Expected behavior: The function Mcl_DmaGetCrtIterCount only get CITER bits in Current major iteration count. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update function Mcl_IPW_DmaGetChCiter (It can be similar to Mcl_IPW_DmaGetChBiter). Check all iter functions if the implementation is correct: Mcl_DmaTcdSetIterCount Mcl_DmaTcdSetLinkAndIterCount Mcl_DmaTcdGetIterCount Mcl_DmaUpdateIterCount Mcl_DmaGetCrtIterCount Mcl_DmaGetStartIterCount",'Update function Mcl_DmaGetCrtIterCount
ENGR00372105	NewWork	[MCL] Update the DEM reporting 'NewWork Description: The DEM error codes are separate code structures.They should be moved in the MCL init config pointer. The DMA must be updated accordingly(to use new path of the codes when reporting to DEM) Expected behavior: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Move DEM error code structures into the MCL init configuration structure. Update paths for DEM error reporting calls.","'The DEM error codes are separate code structures.They should be moved in the MCL init config pointer. The DMA must be updated accordingly(to use new path of the codes when reporting to DEM)",ENGR00372107
ENGR00370087	NewWork	[MCL] Updates needed to support ASR 4.2.1 and ASR 4.0.3 in the smcal code base 'NewWork Description: Updates needed to support ASR 4.2.1 and ASR 4.0.3 in the smcal code base. See attached checklist.",'Updates needed to support ASR 4.2.1 and ASR 4.0.3 in the smcal code base
ENGR00374014	NewWork	[MCL] Use "READONLY" instead of "EDITABLE" in xdm file 'Description 1) Nodes that have constant false editable conditions (<a:a name="EDITABLE" value="false"/>) will change to (<a:a name="READONLY" value="true"/>) 2) Exception: the 3 nodes in container <Mdl>_ModuleDescription must continue to have <a:a name="EDITABLE" value="false"/>

Table continues on the next page...

ID	Subtype	Headline and Description
		3) Nodes that have EDITABLE attributes with type xPath will not be changed to READONLY","'- Use "READONLY" instead of "EDITABLE" in xdm file
ENGR00378359	NewWork	[MCL] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Analyze all existing MISRA comments and fix all of them. Cleanup unused MISRA comments. Exceptions which are accepted for the MISRA deviations are present in the attached excel file.",'Fix and cleanup MISRA errors.,ENGR00379091 ENGR00379092
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)(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. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: DONE bit isn't cleared when using the function Mcl_IPW_DmaClearDone for S32K Expected behavior: DONE bit is cleared on both of platfrom with offset 0x1f and 0x1C 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
		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));","'File modified: - Mcl_IPW.c
ENGR00373800	NewWork	[MCL]Update the changes from Racerunner Ultra for ASR 4.0.3 'NewWork Description: This ticket will aggregate the changes in RR Ultra for ASR 4.0.3",'Update the changes from Racerunner Ultra for ASR 4.0.3
ENGR00369959	Defect	[MCU] Ambiguous configuration parameter description McuFXOSC_OSCM 'Problem detailed description (how to reproduce it): The following description is not cleared Name: McuFXOSC_OSCM Crystal Oscillator Mode - FXOSC_CTL[OSCM]. This bit selects the oscillator mode when not in bypass. Value Description 0 FSP mode (full swing pierce mode) 1 LCP mode (loop controlled pierce mode It should be something like: Name: McuFXOSC_OSCM Crystal Oscillator Mode - FXOSC_CTL[OSCM]. This bit selects the oscillator mode when not in bypass. Value Description Checked: FSP mode (full swing pierce mode) Uncheck: LCP mode (loop controlled pierce mode Proposed solution (Optional): Update description of the field to indicate what each state of the checkbox will generate and update RegOperations.m according to the description and the reference manual.","'Update description of the field to indicate what each state of the checkbox will generate and update RegOperations.m according to the description and the reference manual. Mcu.xdm Mcu_RegOperations.m
ENGR00369776	Defect	[MCU] BAF_DIS not configurable for Halo 'Problem detailed description (how to reproduce it): For the version of IPV_FLASHV2 present on Halo the masks for PFLASH_PFCR3[BAF_DIS] are not present Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		N/A Trigger: N/A Observed behavior: BAF_DIS not configurable for Halo Expected behavior: BAF_DIS configurable for Halo 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 Mcu_FLASH_Types.h, add following code: #if ((IPV_FLASHV2 == IPV_FLASH_40_96_55_19) #define MCU_FLASH_BAF_DIS_U32 (FLASHC_PFCR3_BAF_DIS_MASK32) #define MCU_FLASH_BAF_EN_U32 (0x00000000U) #endif","modified files: Mcu_FLASH_Types.h
ENGR00380511	Defect	[MCU] Compiler warning in template files when MCU_INIT_CLOCK == STD_OFF 'Problem detailed description (how to reproduce it): EMIOS configuration structure is only used when MCU_INIT_CLOCK == STD_ON. It should be guarded by #if (MCU_INIT_CLOCK == STD_ON) to avoid the following compiler warnings: warning #177-D: variable "EMIOS_Module_ConfigPC0_0" was declared but never referenced@48 warning #177-D: variable "EMIOS_Module_ConfigPB0_0" was declared but never referenced@48 Preconditions: MCU_INIT_CLOCK == STD_OFF Build test case Mcu_TS_M06 Test Case ID (internal TC that caught the defect) - optional Mcu_TS_M06 Trigger: N/A Observed behavior: N/A Expected behavior: No warning appear in report Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Guard EMIOS configuration by MCU_INIT_CLOCK == STD_ON.","- Mcu.xdm - Mcu_RegOperations.m
ENGR00381326	Defect	[MCU] Correct misra errors for Calypso RTM 1.0.2 'Problem detailed description (how to reproduce it): Misra Rule 8.10 errors detected in Mcu_FLASH.c files: In the implement of CR ENGR00357962, The call Mcu_FLASH_Init was guarded by (MCU_DISABLE_FLASH_CONFIG == STD_OFF), but the Mcu_FLASH_Init function wasn't guarded by (MCU_DISABLE_FLASH_CONFIG == STD_OFF). This is raised a misra errors. The same for Mcu_FLASH_SetWS

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: MCU_DISABLE_FLASH_CONFIG == STD_ON Test Case ID (internal TC that caught the defect) - optional Mcu_TS_014 Trigger: MCU_DISABLE_FLASH_CONFIG == STD_ON Observed behavior: Have misra errors Expected behavior: No misra errors Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Mcu_FLASH_Init and Mcu_FLASH_SetWS need be guarded by (MCU_DISABLE_FLASH_CONFIG == STD_OFF)","'Mcu_FLASH.c Mcu_FLASH.h
ENGR00375720	NewWork	[MCU] Fix Compiler waring and Misra warnings 'NewWork Description: In Mcu, we got this warning as below: warning #826-D: parameter "RamSection" was never referenced@90 FUNC(Std_ReturnType, MCU_CODE) Mcu_InitRamSection(VAR(Mcu_RamSectionType, AUTOMATIC) RamSection) warning #549-D: variable "RamStatus" is used before its value is set@29 return (Std_ReturnType)RamStatus; Proposal solution: Find and fix Compiler/misra warning",'Fix compiler warning
ENGR00380815	NewWork	[MCU] Generation of Mcu via command line always delivers a huge amount of warnings regarding McuResetReason containers 'NewWork Description: WARNING 16-07-15,10:18:15 (1134) Cannot set the value of the node / AUTOSAR/TOP-LEVEL-PACKAGES/ActiveEcuC/ELEMENTS/Mcu/ McuPublishedInformation/McuResetReasonConf/MCU_1_2_LV_RESET/ McuResetReason to 4 because it is readonly. Its DEFAULT is used instead. This message is thrown for each configured McuResetReason and occurs during generation of Mcu and all other components that depend from Mcu. This might result in some hundreds of warnings. Therefore we already received complaints of some customers. Suppressing those messages (specific warning ID via additional parameter when calling Tresos) would be possible, but it is no generic solution and does not consider the root cause of the problem. You can easily reproduce this issue via command line (for MPC574XG_MCAL4_0_RTM_1_0_1). When does this happen: ---------- At generation time. In which configuration does this happen: ---------- Any. Discussion: ---------- Other vendors using Tresos as generation tool already provided a solution via

Table continues on the next page...

ID	Subtype	Headline and Description
		adaptation of Mcu.xdm (default value of parameter McuResetReason; find attached as McuResetReason_Patch.xml). The Mcu plugin contains McuPreConfiguration.xdm (Pre configuration of McuResetReason containers); this is Tresos proprietary format. We expect to find the according pre configuration additionally in AUTOSAR format as part of the MCAL package as an assistance for the user. Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update McuResetReason in Mcu.xdm to add the values of the ResetReasons in the DEFAULT tag to avoid warnings when generating code.",'Update McuResetReason in Mcu.xdm to add the values of the ResetReasons in the DEFAULT tag to avoid warnings when generating code.
ENGR00376396	Defect	[MCU] Guard macro for Reg_eSys_STCU.h file is incorrect 'Problem detailed description (how to reproduce it): Guard macro for Reg_eSys_STCU.h is incorrect. #ifndef REG_ESYS_STCU_H #define Reg_ESYS_STCU_H Preconditions: IPV_STCU2 is used in the code Test Case ID (internal TC that caught the defect) - optional Found via MISRA review Trigger: File inclusion problem Observed behavior: The file is not protected from multiple inclusions Expected behavior: The file should be protected from multiple inclusions Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Replace in Reg_eSys_STCU.h #ifndef REG_ESYS_STCU_H #define Reg_ESYS_STCU_H with #ifndef REG_ESYS_STCU_H #define REG_ESYS_STCU_H","'Fix guard macro. Updated files: Reg_eSys_STCU.h
ENGR00380285	NewWork	[MCU] 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 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.

Table continues on the next page...

ID	Subtype	Headline and Description
		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: Planned requirement (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fill in the attached template Exclusive areas defined in driver code",'Analyze data consistency
ENGR00368157	NewWork	[MCU] Improve Dem/Det error description in Usermanual 'NewWork Description: Please analysis to improve Usermanual with description of Dem/Det errors, that

Table continues on the next page...

ID	Subtype	Headline and Description
		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. Requirement source: (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Dem/Det error description in Usermanual should be improve to more clear for user We can count on the MCU Specification to update more detail for UM",'Update to UM document
ENGR00366803	NewWork	[MCU] Improve McuClockReferencePoint definition 'NewWork Description: In the McuClockReferencePoint definition, it would be easier (less error-prone) to have EMIOS_0, _1, and _2 as well in the drop-down list for parameter McuClockFrequencySelect. It could be calculated automatically by the tool, instead of having to define a "CUSTOM" clock. Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add EMIOS_CLK as a clock reference point",'Add EMIOS_CLK as a clock reference point
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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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];","'modified files: Mcu.c Mcu_IPW.c",ENGR00369926 ENGR00370153 ENGR00370803
ENGR00365732	Defect	[MCU] Incorrect definition for MCU_FLASH_MxAP_U32 'Problem detailed description (how to reproduce it): PFLASH MXAP are always written with their mask value because MCU_FLASH_MxAP_U32 macros are defined incorrectly in the Mcu_FLASH_Types.h file. Even if configured with 0, every bit from PFLASH_PFAPR will be written with 1. Problem detailed description (how to reproduce it): This CR appeared as a result of the incomplete implementation of ENGR00373040 in an internal release. Preconditions: Trying to configure any bit in PFLASH_PFAPR with 0. Test Case ID (internal TC that caught the defect) - optional NA Trigger: Calling Mcu_Init and trying to configure any PFLASH_PFAPR bit with 0. Observed behavior: PFLASH_PFAPR is always written with 0xFFFF_FFFF Expected behavior: PFLASH_PFAPR should be written with what was configured 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 macros for MCU_FLASH_MxAP_U32 in Mcu_FLASH_Types.h Replace #define MCU_FLASH_MxAP_U32(value) ((uint32)((uint32)((value) << (uint32)xU) | FLASHC_PFAPR_MxAP_MASK32))

Table continues on the next page...

ID	Subtype	Headline and Description
		with #define MCU_FLASH_MxAP_U32(value) ((uint32)((uint32)((value) << (uint32)xU) & FLASHC_PFAPR_MxAP_MASK32))","'Change #define MCU_FLASH_MxAP_U32(value) ((uint32)((uint32)((value) << (uint32)xU) | FLASHC_PFAPR_MxAP_MASK32)) to #define MCU_FLASH_MxAP_U32(value) ((uint32)((uint32)((value) << (uint32)xU) & FLASHC_PFAPR_MxAP_MASK32))
ENGR00372019	Defect	[MCU] Incorrect shift value of SSCM_MEMCONFIG REG 'Problem detailed description (how to reproduce it): In file Mcu_SSCM.c instead of u32State |= (uint32((uint32)REG_READ16(SSCM_MEMCONFIG_ADDR16) << (uint32)0x16U); it should be u32State |= (uint32((uint32)REG_READ16(SSCM_MEMCONFIG_ADDR16) << (uint32)0x10U); Preconditions: MCU_GET_SYSTEM_STATE_API == STD_ON Test Case ID (internal TC that caught the defect) - optional Mcu_TC_0152 Trigger: MCU_GET_SYSTEM_STATE_API == STD_ON Observed behavior: Value of Mcu_SSCM_GetSystemState incorect Expected behavior: Value of Mcu_SSCM_GetSystemState corect Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): line 273, file Mcu_SSCM.c.This line should be: u32State |= (uint32((uint32)REG_READ16(SSCM_MEMCONFIG_ADDR16) << (uint32)0x10U);","'Modified files: Mcu_SSCM.c
ENGR00371368	Defect	[MCU] LVD_PD2_COLD_REE bit and LVD_IO_HI_RE are not set 'Problem detailed description (how to reproduce it): The mask value of PMCDIG_MCR is not correct. In RTM1.0.1: #define PMCDIG_MCR_RWBITS_MASK32 ((uint32)0x00008680U) But according to Reference Manual Rev4. It should be: #define PMCDIG_MCR_RWBITS_MASK32 ((uint32)0x00008D00U) Because of the wrong mask value, the bit LVD_PD2_COLD_REE and LVD_IO_HI_RE are not set even the customer has enabled the bits under EB Tresos. Preconditions: McuPmcMcr_LvdPd2ColdRee and McuPmcMcr_LvdIoHiRee are checked under EB Tresos. Observed behavior: LVD_PD2_COLD_REE bit and LVD_IO_HI_RE are not set Expected behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		LVD_PD2_COLD_REE bit and LVD_IO_HI_RE are set Note: Reference Manual Rev4. Proposed solution (Optional): Correct PMCDIG_MCR_RWBITS_MASK32 value.",'Fix PMCDIG_MCR_RWBITS_MASK32
ENGR00373040	Defect	[MCU] MCU does not configure all PRAM controllers 'Problem detailed description (how to reproduce it): On Calypso3M (including mpc5746c devices) there are 2 banks of system SRAM with dedicated Platform RAM Controller (PRAMC) modules. In the current implementation the register that configures the platform RAM Controller for the second SRAM bank (PRAMC1_PRCR1) is not configured in the xdm and its value will always be the default one because it is not written in Mcu_PRAM.c. As a consequence PRAMC_1_PRCR1[FT_DIS] will always be 0. Preconditions: 1. "#define MCU_PRAM_MULTIPLE_PRAM STD_OFF" in Mcu_Cfg.h 2. "#define MCU_SRAMC_SUPPORT (STD_ON)" in Mcu_Cfg.h 3. "#define IPV_MC (0x05000007UL) " is fixed according to Soc_Ips.h Test Case ID (internal TC that caught the defect) - optional NA Trigger: Call Mcu_InitClock Observed behavior: PRAM1_PRCR1 is not configured Expected behavior: PRAM1_PRCR1 should be configured Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Configure all PRAM controllers by updating the SRAMC_ClockConfig in Mcu_RegOperations.m and updating in the xdm the editable tag of FT_DIS for PRAM1_PRCR1 note that MCU.MultiplePRAM = true, it means that 3 PRAM controllers MCU.MultiplePRAM = false, it means that 2 PRAM controllers",'Configure all PRAM controllers,ENGR00373211
ENGR00365239	Defect	[MCU] MCU driver does not check if it is initialized, when and interrupt is generated 'Problem detailed description (how to reproduce it): The Mcu driver is not sastify with: [SMCAL_SW035] ISR shall check whether its respective driver is initialized. If the driver is not initialized, the ISR shall only clear interrupt status flag and return Preconditions: Interrupts appear before the driver is initialized Test Case ID (internal TC that caught the defect) - optional NA Trigger: Interrupts appear before the driver is initialized Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		The configuration piointer is not checked for being NULL Expected behavior: The configuration pointer should be checked for being NULL, and if it is the interrupt should just clear the flags Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Check cfg_ptr with NULL_PTR before handler isr","'Mcu.c Mcu.h",ENGR00381086 ENGR00381088 ENGR00381089 ENGR00381090
ENGR00357962	Defect	[MCU] MCU-Driver causes code execution from RAM 'The MCU-Driver contains two functions. * Mcu_FLASH_init_Ram() * Mcu_FLASH_SetWS_Ram() ... which require code execution from RAM. This feature does not work under all conditions because it's blocked by HW. Request: * These functions should be disabled in the next release or modified so that they lead to an error response. Alternatively the functions get updates so that they can only be executed by the HSM core. Customer updated: In this activity I would like to outline the discussions we had on the topic. 1. Documentation. Looking at Calypso 3M Rev. 1 Reference Manual - 9.5.1 Flash memory controller: NOTE PFLASH_PFCR1[APC] and PFLASH_PFCR1[RWSC] field values needs to be updated while read accesses to flash are stopped, that is, the software code which updates the PFLASH_PFCR1[APC] and PFLASH_PFCR1[RWSC] fields must be executed from RAM. Was this requirement reworked for Castor? Seems not, because Custor RM in most cases refers to the Calypso RM. Anyway for the Castor we need to work out a way of handling of the configuration of the Flash Wait States. 2. Customer concerns. First, The customer wanted to know - what that function does? Mihai explained: - the Mcu_FLASH_init_Ram() is a function from the IPV layer, a low level function that directly accesses the hardware registers (specifically PFLASH_PFCR, except the bit fields that contain the wait state for the flash). It is used as part of the Mcu_Init() AUTOSAR API. - the Mcu_FLASH_SetWS_Ram() configures the wait state for the flash. It is used as part of Mcu_InitClock(). Second, once there are devices where execution from RAM is disabled, we need the following: - ensure the RM does not mislead Designer by demanding "config code execution from RAM" - need another method to configure wait states and other flash parameters The solution for this proposed: 1) configure the flash inside HSM, which allows execution from RAM 2) from cache",'Add a checkbox in the plugin that will generate a precompile switch that will allow to choose not to use the Flash functions that are executed from RAM

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00376879	NewWork	[MCU] MCU_CMU_BASE_ADDR extern declaration can be removed from Reg_eSys_CMU.h 'NewWork Description: Remove MCU_CMU_BASE_ADDR extern declaration from Reg_eSys_CMU.h to avoid 2 misra errors. Requirement source: Internal refactoring (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Remove MCU_CMU_BASE_ADDR extern declaration from Reg_eSys_CMU.h.","'Remove MCU_CMU_BASE_ADDR extern declaration from Reg_eSys_CMU.h to avoid 2 misra errors. File updated: Reg_eSys_CMU.h
ENGR00363249	Defect	[MCU] MCU_LVD_LV_PD2_COLD_RESET event missing from McuPreConfiguration.xdm 'Problem detailed description (how to reproduce it): MCU_LVD_LV_PD2_COLD_RESET event missing from McuPreConfiguration.xdm Preconditions: Mcu_GetResetReason or Mcu_GetResetRawReason are used and MCU_LVD_LV_PD2_COLD_RESET is detected Test Case ID (internal TC that caught the defect) - optional NA Trigger: Mcu_GetResetReason or Mcu_GetResetRawReason are used and MCU_LVD_LV_PD2_COLD_RESET is detected Observed behavior: MCU_LVD_LV_PD2_COLD_RESET is missing from McuPreConfiguration.xdm Expected behavior: MCU_LVD_LV_PD2_COLD_RESET should be in McuPreConfiguration.xdm Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add MCU_LVD_LV_PD2_COLD_RESET event in McuPreConfiguration.xdm",'MCU_LVD_LV_PD2_COLD_RESET event missing from McuPreConfiguration.xdm
ENGR00381030	NewWork	[MCU] MCU_SSCM_ERROR_PRESENT is not defined on all the versions of the IP 'NewWork Description: MCU_SSCM_ERROR_PRESENT is not defined on all the versions of the IP and it causes the LDRA build to fail Requirement source: Internal refactoring (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
		Only check MCU_SSCM_ERROR_PRESENT on IPV_SSCM_06_00_00_06 version of SSCM','Only check MCU_SSCM_ERROR_PRESENT on IPV_SSCM_06_00_00_06 version of SSCM
ENGR00368690	Defect	[MCU] McuExternalTimeBase should be gray out (unconfigurable) 'Problem detailed description (how to reproduce it): Currently, McuExternalTimeBase is configurable, ETB will be written 0 or 1 depend on user configuration. According to MPC574XG RMv4, EMIOS_MCR[ETB] should always 0 since 1 is reserved. Therefore McuExternalTimeBase should be gray out (unconfigurable) Preconditions: McuEMIOSControl is checked to configure EMIOS using Mcu_InitClock Test Case ID (internal TC that caught the defect) - optional NA Trigger: Configuration problem in the xdm, triggered by checking McuEMIOSControl and configuring McuExternalTimeBase Observed behavior: McuExternalTimeBase can be both checked and unchecked with no restrictions Expected behavior: McuExternalTimeBase should be not editable Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): McuExternalTimeBase should be gray out (unconfigurable)","'McuExternalTimeBase should be gray out (unconfigurable) modified files: Mcu.xdm
ENGR00362609	Defect	[MCU] Mcu_GetResetRawValue() not getting all reset reasons 'Problem detailed description (how to reproduce it): The raw value returned by Mcu_MC_RGM_GetResetRawValue() does not contain all the reset bits reviously set in FES register. The reset reason bit returned by Mcu_MC_RGM_GetResetReason() is missing. Preconditions: -If no bit are set in DES register and more than one bit is set in the FES register (for example ST_DONE and EXR), -If Mcu_MC_RGM_GetResetReason() and then Mcu_MC_RGM_GetResetRawValue() are called, -The raw value returned by Mcu_MC_RGM_GetResetRawValue() does not contain all the reset bits previously set in FES register. The reset reason bit returned by Mcu_MC_RGM_GetResetReason() is missing. Test Case ID (internal TC that caught the defect) - optional n/a Trigger: Mcu_MC_RGM_GetResetReason() or then Mcu_MC_RGM_GetResetRawValue() are called Observed behavior: Mcu_GetResetRawValue() get all reset reasons Expected behavior: get all and right reset reason.

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): Please get more details in attachment file.","'Update Mcu_GetResetReason() according to ASR 4.2.1. It should return a same reset reason every time and clear all reset reasons once flag's read. In addition, make sure that all reset reasons of DES/FES registers are stored in global variables, that would be used by Mcu_GetResetRawValue() to get raw value. UPDATE more requirement of GetResetReason(): 1) Add a MCU_MULTIPLE_RESET_REASON to the enumeration in case more than one bit(except POR) are set in FES and DES so the application will know that it should call GetResetRawValue() next after calling GetResetRawReason(). 2) Add a MCU_NO_RESET_REASON in case both FES and DES are 0x00 and the global variables are also 0x00. This may happen on Calypso for example if all bits are cleared by a GetResetReason call and then a wakeup happens. The wakeup clears the variables and doesn't set any flags, so the function will not know what to return.",ENGR00364139
ENGR00380914	Defect	[MCU] Mcu_Ipw_RGM_ClearClockFailIrqFlags function was renamed, but the call was not updated 'Problem detailed description (how to reproduce it): The name of Mcu_Ipw_RGM_ClearClockFailIrqFlags function is incorrect. It should be Mcu_Ipw_RgmClearClockFailIrqFlags. This is due to incomplete implementation of the code review against checklist. This issue affects an internal release of one of the MCU driver components. Does not affect the existing Calypso RTM and will be fixed by the next RTM. For Calypso platform, the issue not present on customer side. Preconditions: Call: Mcu_Cmu_ClockFailIsr Test Case ID (internal TC that caught the defect) - optional N/A Trigger: (MCU_RGM_CLEAR_CLOCK_IRQ_FLAG == STD_ON) Observed behavior: Build fail 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): Correct name of Mcu_Ipw_RGM_ClearClockFailIrqFlags function. It should be Mcu_Ipw_RgmClearClockFailIrqFlags.",'Corrected the Define Mcu_Ipw_CmuClearClockFailIrqFlags in Mcu_CMU.c
ENGR00366776	Defect	[MCU] Missing MemMap declarations in driver files 'Problem detailed description (how to reproduce it): The following variables do not have consistent memory section guarding for definition and all other extern declarations: Eth_InternalCfgPtr Fls_ACEraseRomEnd

Table continues on the next page...

ID	Subtype	Headline and Description
		Fr_FrDemCtrlTestResultPtr Lin_Config_PC Mcu_Config_PC Port_pConfig Wdg_au8Index Wdg_pConfigPC the variables shoudl be allocted in the same MemMap.h section where it is defined, and in every other place where it's declared external. this causes a link error when sda=all compiler option is used. Preconditions: No special precondition. Compiler/linker problem. Test Case ID (internal TC that caught the defect) - optional NA Trigger: Compiling the code Observed behavior: Linker error. Expected behavior: No errors. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add Mcu_Config_PC in the right memory map section",'Add MemMap declarations in Mcu.c,ENGR00366981
ENGR00380362	Defect	[MCU] Node McuCmuNotification() is not checked for being NULL_PTR in Mcu.xdm 'Problem detailed description (how to reproduce it): Node McuCmuNotification() is not checked for being NULL_PTR in Mcu.xdm. The value NULL_PTR for the node should not be acceptable. Preconditions: Enable node McuCmuNotification() Test Case ID (internal TC that caught the defect) - optional review Trigger: Enable node McuCmuNotification() Observed behavior: 'NULL_PTR' is accepted Expected behavior: 'NULL_PTR' is not accepted Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Add condition to check that the value of this node is not NULL_PTR",'Add to check node McuCmuNotification()
ENGR00381263	Defect	[MCU] PFLASH_PFCR3[P2_WCFG] was not be configure 'Problem detailed description (how to reproduce it): PFLASH_PFCR3[P2_WCFG] was not configured by driver Preconditions: Call Mcu_Init and check value of PFLASH_PFCR3 register Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		Any Trigger: P2_WCFG bit is configured different 0 value Observed behavior: PFLASH_PFCR3[P2_WCFG] was not configured by driver Expected behavior: PFLASH_PFCR3[P2_WCFG] was configured by driver Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Check value of McuPort2PageBufferConfiguration to configure for P2_WCFG bits",'Mcu_RegOperations.m
ENGR00378453	Defect	[MCU] PRAMC_PRCR1_RESBITS_MASK32 macro is incorrect 'Problem detailed description (how to reproduce it): [IPV_PRAM] PRAMC_PRCR1_RESBITS_MASK32 macro is incorrect Preconditions: IPV_MC == IPV_MC_05_00_00_07 Test Case ID (internal TC that caught the defect) - optional NA Trigger: Call Mcu_InitClock Observed behavior: PRAMC_PRCR1 is not configured Expected behavior: PRAMC_PRCR1 should be configured Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Follow RM: PRAMC_PRCR1_RESBITS_MASK32 macro should be 0x00000200U","'PRAMC_PRCR1_RESBITS_MASK32 macro is incorrect - Reg_eSys_PRAM.h
ENGR00372697	Defect	[MCU] Replace hard-coded copyright to M4_SRC_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Some plugin files contain hard-coded copyright Preconditions: No special preconditions Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: In some plugin files the copyright is hardcoded Expected behavior: In all driver files the copyright shall be M4_SRC_COPYRIGHTED_TO to be replaced when building the plugin with the correct information. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Replace hard-coded copyright to M4_SRC_COPYRIGHTED_TO",'Replace

Table continues on the next page...

ID	Subtype	Headline and Description
		hard-coded copyright to M4_SRC_COPYRIGHTED_TO
ENGR00379534	NewWork	[MCU] Review and implement changes according to Reference Manual for Calypso 3M/6M RTM 1.0.2 'NewWork Description: - Review and implement changes according to the reference manual for Calypso 3M/6M RTM 1.0.2 provided in the attachment. - Update Release note field to "Y" if you change anything in code/UM/IM according to this review. Otherwise, leave it as "N". Requirement source: - RM for Calypso 3M: MPC5746C_RM_Rev3.pdf, Rev. 3, 12/2015 http://compass.freescale.net/go/231236725 - RM for Calypso 6M: MPC5748G Reference Manual, Rev. 4, 07/2015 http://compass.freescale.net/go/227152351 (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Analyzing","'Mcu_mpc574xx_xxx.txt Mcu.xml Mcu_RegOperations.m",ENGR00381204
ENGR00364104	Defect	[MCU] Three McuResetReasons are missing in McuPreConfiguration.xdm 'Problem detailed description (how to reproduce it): MCU_NO_RESET_REASON, MCU_MULTIPLE_RESET_REASON are missing from McuPreConfiguration.xdm Preconditions: Mcu_GetResetReason or Mcu_GetResetRawReason are used Test Case ID (internal TC that caught the defect) - optional NA Trigger: Mcu_GetResetReason or Mcu_GetResetRawReason are used Observed behavior: MCU_NO_RESET_REASON, MCU_MULTIPLE_RESET_REASON are present in the code but missing from McuPreConfiguration.xdm Expected behavior: MCU_NO_RESET_REASON, MCU_MULTIPLE_RESET_REASON should be present in McuPreConfiguration.xdm Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update McuPreConfiguration.xdm to add missing Mcu Reset Reasons: changed: <d:ctr name="MCU_RESET_UNDEFINED" type="IDENTIFIABLE"> <d:var name="McuResetReason" type="INTEGER" value="23"> </d:var> </d:ctr> to <d:ctr name="MCU_LVD_LV_PD2_COLD_RESET" type="IDENTIFIABLE"> <d:var name="McuResetReason" type="INTEGER" value="23"> </d:var> </d:ctr> <d:ctr name="MCU_NO_RESET_REASON" type="IDENTIFIABLE">

Table continues on the next page...

ID	Subtype	Headline and Description
		<d:var name="McuResetReason" type="INTEGER" value="24"> </d:var> </d:ctr> <d:ctr name="MCU_MULTIPLE_RESET_REASON" type="IDENTIFIABLE"> <d:var name="McuResetReason" type="INTEGER" value="25"> </d:var> </d:ctr> <d:ctr name="MCU_RESET_UNDEFINED" type="IDENTIFIABLE"> <d:var name="McuResetReason" type="INTEGER" value="26"> </d:var> </d:ctr>","'Modified files: McuPreConfiguration.xdm
ENGR00367155	NewWork	[MCU] Update Mcu_MC_Irq.c to remove Mcu_PMC.h include 'NewWork Description: Update Mcu_MC_Irq.c to remove PMC dependencies Expected behavior: Mcu_PMC.h should not be included on all versions of the IP Requirement source: internal refactoring (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Remove the inclusion of Mcu_PMC.h and include MC_PCU instead",'Add support for both PMC and MC_PCU in MCV4
ENGR00379267	NewWork	[MCU] Update UM with description of timeout functionality and recommendations for configuring timeout parameters 'NewWork Description: Update UM with description and recommendations for configuring timeout parameters. Expected behavior: Update the description for timeout value field in the plugin to include a recommended value Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Add a recommended value for the timeout value from the plugin","'Update UM to describe the timeout values recommended for the client. See attached excel for Magic Carpet recommended timeout
ENGR00381516	NewWork	[MCU] Update compiler warning according to the CR ENGR00381326 'Problem detailed description (how to reproduce it): The warning was detected by Mcu_TS_Eq_Cot_01 warning #177-D: function "Mcu_FLASH_InitRam" was declared but never referenced@34 warning #177-D: function "Mcu_FLASH_SetWsRam" was declared but never referenced@34

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: MCU_DISABLE_FLASH_CONFIG == STD_ON Test Case ID (internal TC that caught the defect) - optional Mcu_TS_Eq_Cot_01 Trigger: MCU_DISABLE_FLASH_CONFIG == STD_ON Observed behavior: N/A Expected behavior: No warning appear in report Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Mcu_FLASH_InitRam and Mcu_FLASH_SetWsRam need be guarded MCU_DISABLE_FLASH_CONFIG == STD_OFF",'Mcu_FLASH.c,ENGR00381520
ENGR00373770	NewWork	[MCU] Use "READONLY" instead of "EDITABLE" in xdm file for EDITABLE set to false 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: Quality (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Replace EDITABLE tags with READONLY tags for EDITABLE set to false except the ModuleDescription choice container Note: The logic of the conditions must be reversed.",'Replace EDITABLE with READONLY tags in xdm
ENGR00378353	NewWork	[MCU] Verify all Misra errors and comments 'NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Verified all MISRA comments for Calypso Rtm 1.0.2",'N/A,ENGR00380800 ENGR00380802 ENGR00380805 ENGR00380807 ENGR00381424 ENGR00381429 ENGR00381432 ENGR00381434 ENGR00381435 ENGR00381436 ENGR00381438 ENGR00381440 ENGR00381443 ENGR00381447 ENGR00381448
ENGR00364235	Defect	[MCU] 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"!][!// [!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:

Table continues on the next page...

ID	Subtype	Headline and Description
		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","'Modified files: Mcu.mak Mcu_Cfg.h Mcu_Cfg.c Mcu_PCfg.c new files added: Mcu_VersionCheck_Inc.m Mcu_VersionCheck_Src_PB.m Mcu_VersionCheck_Src_PC.m
ENGR00365704	Defect	 [MCU] Wrong in usage macro P2FUNC() 'Problem detailed description (how to reproduce it): Wrong parameters in P2FUNC usage in FLASHV2 in the following code: /** @violates @ref Mcu_FLASH_h_REF_2 these object are used trough the entire MCU code*/ extern P2FUNC (void, const, Mcu_FLASH_Init) (P2CONST(Mcu_FLASH_ConfigType, AUTOMATIC, MCU_APPL_DATA) FLASH_pConfigPtr); #if (MCU_INIT_CLOCK == STD_ON) /** @violates @ref Mcu_FLASH_h_REF_2 these object are used trough the entire MCU code*/ extern P2FUNC (void, const, Mcu_FLASH_SetWS) (P2CONST(Mcu_FLASH_ClockConfigType, AUTOMATIC, MCU_APPL_DATA) FLASH_pClockConfigPtr); #endif /* (MCU_INIT_CLOCK == STD_ON) */ Preconditions: No special preconditions Test Case ID (internal TC that caught the defect) - optional NA Trigger: No special trigger Observed behavior: Usage of P2FUNC is incorrect Expected behavior: P2FUNC is used with the correct parameters Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Replace P2FUNC(void, const,...) with P2FUNC(void, MCU_CODE,...)","'Change P2FUNC(void, const,...) to P2FUNC(void,

Table continues on the next page...

ID	Subtype	Headline and Description
		MCU_CODE,...)
ENGR00371384	Defect	[MCU] Wrong label for McuDisableReset in container Mcu_D_LVD_IO_A_HI_ResetSource 'Problem detailed description (how to reproduce it): The label for McuDisableReset in Mcu_D_LVD_IO_A_HI_ResetSource container is incorrect. It shouldn't be Disable TSR_FUNC Event Reset, it should be Disable LVD_IO_A_HI Event Reset Preconditions: No special preconiditions, it is a cosmetic problem in the xdm Test Case ID (internal TC that caught the defect) - optional NA Trigger: No trigger, because it is not a functional defect Observed behavior: The label is incorrect and misleading Expected behavior: The label for the parameter should be Disable LVD_IO_A_HI Event Reset 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 label of the parameter to Disable LVD_IO_A_HI Event Reset","'Correct the description in xdm and UM: files affected: Mcu.xdm The manual UM
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. 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/

Table continues on the next page...

ID	Subtype	Headline and Description
		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 Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Xdm scheme contains duplicated attributes Expected behavior: Remove the redundant ones. 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 redundant ones.','Removed duplicated attributes
ENGR00371861	Defect	[MCU]Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): Incorrect version checking in configuration template files Preconditions: Build test Test Case ID (internal TC that caught the defect) - optional Trigger: Build test Observed behavior: Build failed Expected behavior: Build successful

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): 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 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","'Incorrect version checking in configuration template files Modified files: Mcu_VersionCheck_Inc.m Mcu_VersionCheck_Src_PB.m Mcu_VersionCheck_Src_PC.m
ENGR00372520	NewWork	[PORT] Add IN_OUT support for functionalities marked as i/o which do not have a IMCR setting 'There are a number of functionalities in the IoMuxing excel file of RM rev 4 that are marked as I/O but which do not have a IMCR setting in the Input Muxing excel sheet. For these functionalities, only the IN and OUT signals have been added in the Port_Racerunner.csv. We should also add the IN_OUT signals, with Inmux reg set to 65535 (0xFFFF).','Add IN_OUT support for functionalities marked as i/o which do not have a IMCR setting
ENGR00374772	NewWork	[PORT] Add copyright information in the <platform>_Resource.m files that are generated in the plugin '<platform>_Resource.m files that are generated in the plugin are missing the copyright information. Perl script in the port driver needs to be updated such that a header is added in the <platform>_Resource.m files.','Update Parse.pl
ENGR00378708	Defect	[PORT] Fix code review findings 'Problem detailed description (how to reproduce it): There are some findings after performing review code checklist for Matterhorn RTM 1.0.1 (ENGR00377424) Please see list of findings below: * MisraRule 2.4 - Sections of code should not be "commented out". --> In Parse.pl, there are some lines of code which are commented out

Table continues on the next page...

ID	Subtype	Headline and Description
		* SMCALRule 2.27 - Last line in a file shall be an empty line. --> In Port_VersionCheck_Inc.m, Port_VersionCheck_Src.m and Matterhorn_Resource.m, there is not an empty line at the end of these files * SMCALRule 3.14 --> In Port_Ipw.c, there are lack of parentheses * SMCALRule 5.21, SMCALRule 5.23 --> In Port_Cfg.c and Port_PBCfg.c, there are some macros used the wrong memory allocation Preconditions: NA Test Case ID (internal TC that caught the defect) - optional N/A Trigger: NA Observed behavior: NA Expected behavior: Fix all findings in the attached file. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Please refer to the Proposed Correction column in Detailed_Findings tab of the attached file.",'Fix all findings in the attached file.
ENGR00371380	Defect	[PORT] Fix code review findings against checklist 'Problem detailed description (how to reproduce it): There are some findings after code review against checklist from CR:ENGR00370865 1.Put the end brace of the CONST(Port_Siul2_UnUsedPinConfigType, PORT_CONST) Port_UnUsedPin structure definition in Port_Cfg.c and Port_PBcfg.c on the same column with the start brace. 2. Use the same indentation (same number of spaces) before the lines containing information in <platform>_Resource.m file Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Some of the rules in the checklist for the code review are violated Expected behavior: All applicable rules in the code checklist should be fulfilled 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 Detailed description paragraph above",'Fix findings code review against checklist for RacerunnerIS RTM 2.0.0
ENGR00372956	Defect	[PORT] Fix issue with checking 'PORT_PIN_INOUT' value in macro GetPDDir called from configuration source files 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		Currently, the macro GetPDDir in Port_Operation.m file looks like this: [!IF "text:match(./PortPinDirection,'PORT_PIN_IN')"!][!// [!VAR "MacPinPDDir"="num:i(0)"!][!// [!ELSEIF "text:match(./PortPinDirection,'PORT_PIN_OUT')"!][!// [!VAR "MacPinPDDir"="num:i(1)"!][!// [!ELSEIF "text:match(./PortPinDirection,'PORT_PIN_INOUT')"!][!// [!VAR "MacPinPDDir"="num:i(2)"!][!// [!ENDIF!][!// There is a problem with this approach because the first 'if' matches both PORT_PIN_IN and PORT_PIN_INOUT values. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Configuring direction of a pin to INOUT will result in settings for direction IN to be applied Expected behavior: Configuring direction of a pin to INOUT should result in settings for direction INOUT to be applied Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In order to fix the problem, the last 'elseif' should be moved up, to become the first check.','Fix issue with checking 'PORT_PIN_INOUT' value in macro GetPDDir called from configuration source files
ENGR00363937	NewWork	[PORT] Improve configuration files - Create new macro file 'NewWork Description: A great part of the generated code in Port module is depended on flatform. They are 2 similar configuration files: Port_Cfg.c and Port_PBCfg.c. Currently, if there are problems with them, these files need updated with the same implementation. Therefore, these configuration files can be improved by moving the same parts into a new file containing macros. Expected behavior: There is a file which contains all of macros used by Port_Cfg.c and Port_PBCfg.c Requirement source: It is an improvement (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Create a new file which contains all of macros and configuration files just need call them instead of defining macros 2 times for PB and PC.','Improve configuration files - Create new macro file
ENGR00368573	Defect	[PORT] Inconsistent/Incorrect usage of Compiler Abstraction Keywords and Memory Allocation Sections 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		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];','Review and correct memory allocation,ENGR00378676 ENGR00378682
ENGR00365671	Defect	[PORT] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): Incorrect version checking in configuration template files Preconditions: N/A

Table continues on the next page...

ID	Subtype	Headline and Description
		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): 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 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",'Correct version checking in configuration template files
ENGR00373891	NewWork	[PORT] Optimize the way the information about INOUT pins is stored in the configuration files 'Right now, the information about INOUT pins is stored in the Pad_aFunctExtraSettings[][] array, which is not optimal from required memory point of view. New array should be created, called Port_aPadFunctInoutMuxSettings[] that will store only the information of pads having inout functionality. A new function called Port_Ipw_GetIndexForInoutEntry() should be created in Port_IPW.c file The Perl script Parse.pl should also be updated in order to generate the information in the new Port_aPadFunctInoutMuxSettings[] array","'Update Port_Ipw.c, Parese.pl. Port_Cfg.h, Port_Operation.m to the way the information about INOUT pins is stored in the configuration files
ENGR00373549	Defect	[PORT] PortPinDirection is missing an invalid condition for pins in PortOnlyInputPins list 'For pins that are GPI (only input), Tresos GUI should not allow the user to select direction OUT or IN_OUT. In order to achieve this, a INVALID condition should be added for PortPinDirection attribute in Port_Template.xdm file",'Add one more INVALID

Table continues on the next page...

ID	Subtype	Headline and Description
		condition for PortPinDirection attribute in Port_Template.xdm file
ENGR00373764	Defect	[PORT] Port_SetPinMode() issue when pin>255 due to wrong cast variable to uint8 in Port_Ipw.c 'Problem detailed description (how to reproduce it): in Port_Ipw_SetPinMode(), line 265. Matterhorn RTM 1.0.0 /* Selected word inside the pin description matrix */ u16PinDescWord = (uint16)Port_au16PinDescription[(uint8)PinMode][(uint8)((uint8)PinPad>>4)]; For Pin PS[15] (PCR in MCAL = 272), mode TIM5_7 (PCR = 559), SSS was not changed as expected. Issue is coming from casting variable to uint8 in line code as above: (uint8)PinPad>>4 which is incorrect for pcr >255 ? Preconditions: Port Pin PCR > 255 Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Port_Ipw_SetPinMode() does not work correctly for Pin PCR > 255 Expected behavior: Port_Ipw_SetPinMode() does works correctly for any value of the Pin PCR 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 to uint8 of PinPad from [(uint8)((uint8)PinPad>>4)]",'Remove the wrong cast to uint8 of PinPad variable in Port_Ipw.c.
ENGR00363263	Defect	[PORT] Remove DET error "PORT_E_MODE_UNCHANGEABLE" in Port_Ipw.c 'Problem detailed description (how to reproduce it): With a mode that don't pass statement "if (u16PinDescWord & (uint16)(1UL<<u8PinDescBitOffset))". It will raise only an error "PORT_E_PARAM_INVALID_MODE" 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):","'[PORT] Remove DET error "PORT_E_MODE_UNCHANGEABLE" in Port_Ipw.c
ENGR00377998	NewWork	[PORT] Remove code under define SIUL2_INPUT_WITH_MSCR_SSS_U32 that is no longer used 'Port_Siul2_SetPinMode() function in Port_Siul2.c file contains some code under the define SIUL2_INPUT_WITH_MSCR_SSS_U32 which is not used any more on any of the platforms having SIUL2. Remove this code from the function.,'Remove code in Port_Siul2_SetPinMode() function under the not

Table continues on the next page...

ID	Subtype	Headline and Description
		anymore used define SIUL2_INPUT_WITH_MSCR_SSS_U32
ENGR00365071	Defect	[PORT] Remove compiler warning 'Problem detailed description (how to reproduce it): Compiler warning was reported by some of the compilers used. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Compiler warning was reported by some of the compilers used. Expected behavior: No compiler warnings generated for the driver's code Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Rewrite code in order to remove the compiler warnings or comment them if they cannot be removed.",'Remove the not used variable that generate the compiler warning
ENGR00372702	Defect	[PORT] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: In Port_VersionCheck_Inc.m, Port_VersionCheck_Src.m and Port_VersionCheck_Src_PB.m, there are hard-coded copyright: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics Expected behavior: Plugin files shall not contain hard-coded copyright. Expected behavior: Hard-coded copyright does not exixst in plugin files. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Port_VersionCheck_Inc.m, Port_VersionCheck_Src.m and Port_VersionCheck_Src_PB.m should be updated: From: (c) Copyright M4_SRC_YEAR_ID Freescale Semiconductor Inc. & STMicroelectronics To: (c) Copyright M4_SRC_YEAR_ID M4_SRC_COPYRIGHTED_TO","'Hard-coded copyright need to be replace to M4_SRC_COPYRIGHTED_TO because plugin files shall not contain hard-coded copyright

Table continues on the next page...

ID	Subtype	Headline and Description
		Files Modified: Port_VersionCheck_Inc.m Port_VersionCheck_Src.m Port_VersionCheck_Src_PB.m
ENGR00369180	Defect	[PORT] Update Invalid field for xdm file 'Problem detailed description (how to reproduce it): Invalid field of PortPinPcr, PortPinId, PortNumberOfPortPins is not exactly. Now, Invalid field of PortPinPcr is 0-365 but the max number of PCR is 144 Invalid field PortPinId is 1-366 and PortNumberOfPortPins is 1-500 but the max number of pin is 145 Expected behavior: Invalid field of PortPinPcr is 0-144 Invalid field PortPinId and PortNumberOfPortPins are 1-145 Proposed solution (Optional): Update xdm file","'Update Invalid field for xdm file for PortPinPcr, PortPinId, PortNumberOfPortPins
ENGR00373403	NewWork	[PORT] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.","'Use "READONLY" instead of "EDITABLE" in xdm file
ENGR00378357	NewWork	[PORT] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - Verify all Misra errors and comments - Update the comment of Misra following Comment column in MISRADeviation.xls",'Verify all Misra errors and comments,ENGR00381226
ENGR00365680	NewWork	[PORT]Update Port driver - allow selecting both IN and OUT functionality for a pad defined as I/O 'NewWork Description: Update Port driver - allow selecting both IN and OUT functionality for a pad defined as I/O",'Update Port driver - allow selecting both IN and OUT functionality for a pad defined as I/O
ENGR00381456	Defect	[PORT][DIO] Fix MISRA violation which is reported by new version of MISRA tool 'Problem detailed description (how to reproduce it): In last release, the old tools are not able to report some rule which the content was similar to other rule. For example, in line of code where violate both rule 11.1 and 11.3, only rule 11.1 is reported. When issue in tool is fixed, we have to fix or comment the violations related to these rules. Because of new version of MISRA tool, MISRA rule 11.3 is reported on many lines of Port_Siul2.c and Dio_Siul2.c Preconditions: Casting is used between pointer and integer Test Case ID (internal TC that caught the defect) - optional NA Trigger: Use label tools as below: BLN_CODE-ANALYSIS_01.18.01 BLN_MISRAVIEW_01.09.01 Observed behavior: There are MISRA errors Expected behavior: There should be no MISRA errors unfixed or uncommented. Proposed solution (Optional): Add MISRA comments",'Add comments at the lines where make errors.
ENGR00371916	Defect	[PWM] Correct errorIDs definition 'Problem detailed description (how to reproduce it): Some common errorIDs definition are difference between ASR 4.0.3 and 4.2.1 implementation. In ASR4.0.3:

Table continues on the next page...

ID	Subtype	Headline and Description
		#define PWM_E_PARAM_NOTIFICATION (0x18U) #define PWM_E_PARAM_NOTIFICATION_NULL (0x16U) #define PWM_E_DUTYCYCLE_RANGE (0x17U) #define PWM_E_COUNTERBUS (0x19U) #define PWM_E_CHANNEL_OFFSET_VALUE (0x1AU) #define PWM_E_OPWMB_CHANNEL_OFFSET_DUTYCYCLE_RANGE (0x1BU) #define PWM_E_PARAM_INSTANCE (0x1DU) #define PWM_E_OPWMT_CHANNEL_TRIGGER_RANGE (0x1CU) #define PWM_E_OUTPUT_STATE (0x1EU) #define PWM_E_UNEXPECTED_ISR (0x30U) In ASR4.2.1 #define PWM_E_PARAM_NOTIFICATION (0x30U) #define PWM_E_PARAM_NOTIFICATION_NULL (0x31U) #define PWM_E_DUTYCYCLE_RANGE (0x32U) #define PWM_E_COUNTERBUS (0x33U) #define PWM_E_CHANNEL_OFFSET_VALUE (0x34U) #define PWM_E_OPWMB_CHANNEL_OFFSET_DUTYCYCLE_RANGE (0x35U) #define PWM_E_PARAM_INSTANCE (0x36U) #define PWM_E_OPWMT_CHANNEL_TRIGGER_RANGE (0x37U) #define PWM_E_OUTPUT_STATE (0x38U) #define PWM_E_UNEXPECTED_ISR (0x39U) Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The common errorIDs definition is not same ASR4.0.3 and ASR4.2.1 Expected behavior: The common errorIDs definition should be same ASR4.0.3 and ASR4.2.1 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Pwm.h
ENGR00380154	Defect	[PWM] EMIOS_x_ISR_USED defines should be only generated for channels which use notification 'Problem detailed description (how to reproduce it): At the moment, EMIOS_x_ISR_USED are generated for all cases, if a GPT channel appears, disregarding the actual use of ISR's. This might lead to unnecessary issues in checks when interrupts are shared: #if ((defined ICU_EMIOS_0_CH_1_ISR_USED) && (defined GPT_EMIOS_0_CH_1_ISR_USED)) #error "ICU and GPT resource conflict for EMIOS unified channel EMIOS_0_1" #endif Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Whenever an EMIOS channel is configured, no matter the channel configures

Table continues on the next page...

ID	Subtype	Headline and Description
		notification or not, EMIOS_x_ISR_USED define is always generated. Expected behavior: EMIOS_x_ISR_USED define should only be generated when the channel configures notification. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): 1. In order to avoid resoure conflicts among ICU, GPT and PWM when an EMIOS channel is configured for two or more modules: + GPT_EMIOS_x_USED defines will be generated if an EMIOS channel is configured to use in Gpt + PWM_EMIOS_x_USED defines will be generated if an EMIOS channel is configured to use in Pwm Then, the resoure conflicts check should be: #if ((defined ICU_EMIOS_0_CH_1_ISR_USED) && (defined GPT_EMIOS_0_CH_1_USED)) #error "ICU and GPT resource conflict for EMIOS unified channel EMIOS_0_1" #endif 2. GPT_EMIOS_x_ISR_USED defines should only be generated if the eMIOS channel configures ISR's in Gpt 2. PWM_EMIOS_x_ISR_USED defines should only be generated if the eMIOS channel configures ISR's in Pwm",'Pwm_Cfg.h
ENGR00378693	Defect	[PWM] Fix compiler warning 'Problem detailed description (how to reproduce it): There are some compiler warnings when in the PWM module. The list is: 1. Pwm.c - (dcc:1711): undefined identifier 'PWM_SELECT_COMMON_TIMEBASE_API' used in constant expression - (dcc:1604): Useless assignment to variable u8InvalidParam. Assigned value not used. 2. Pwm.h - (dcc:1711): undefined identifier 'PWM_SELECT_COMMON_TIMEBASE_API' used in constant expression 3. Pwm_Ipw.c - (dcc:1604): Useless assignment to variable retVal. Assigned value not used. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: N/A Expected behavior: The issue need to be fixed in the PWM module 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",'Fixed compile warning,ENGR00378697
ENGR00380547	Defect	[PWM] Fix compiler warnings

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): Due to wrong configuration on ATE, we got the correct CWE reports very late and they contain with some compiler warnings that are needed to be verified and fixed. The following warnings should be fixed in PWM in file Pwm_eMios.c: line 3583 - warning #550-D: variable "peMiosChannelConfig" was set but never used@68 line 3588 - warning #550-D: variable "peMiosChannelConfig" was set but never used@68 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: There are 2 compiler warnings Expected behavior: Compiler warnings should disappear Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Pwm_Ipw.c,ENGR00381126
ENGR00381477	Defect	[PWM] Fix misra errors 'Problem detailed description (how to reproduce it): Fix MISRA errors reported by tool. The new rule that is reported is MISRA 11.3 - A cast should not be performed between a pointer type and a integral type. Preconditions: The driver uses reg operations (REG_WRITE32/REG_BIT_CLEAR32...etc) Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Misra errors exist Expected behavior: There should not be any unregistered misra errors. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Comment all reg operations with MISRA 11.3, because this cannot be fixed.",'Pwm_eMios.c
ENGR00376543	Defect	[PWM] Fix the definition error of master mode selection 'Problem detailed description (how to reproduce it): Error:MASTER_MODE_UP_DOWN_BUFFERD_COUNTER was defined incorrect (Missing " E " letter). Preconditions: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Error:MASTER_MODE_UP_DOWN_BUFFERD_COUNTER was defined incorrect (Missing " E " letter). Expected behavior: Error is fixed: MASTER_MODE_UP_DOWN_BUFFERD_COUNTER was defined incorrect (Missing " E " letter). Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): You should re-define as MASTER_MODE_UP_DOWN_BUFFERED_COUNTER","'Pwm_Cfg.c Pwm_PBcfg.c",ENGR00376544 ENGR00376548
ENGR00378056	Defect	[PWM] Fix variable naming 'Problem detailed description (how to reproduce it): In Pwm_Cfg.h, there is a variable named "w" which " is not compliant with coding style (all variables should have a meaningful name) New name should be "SlashCount" Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: there is a variable named "w" which " is not compliant with coding style (all variables should have a meaningful name) Expected behavior: rename this variable to compliant with coding style Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Pwm_Cfg.h
ENGR00379954	NewWork	[PWM] 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 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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): NA",'Pwm_eMios.c
ENGR00379344	NewWork	[PWM] Implement findings from code review against checklist for Matterhorn RTM 1.0.1 'o Implement findings Checklist is attached to the umbrella ticket and available at: http://compass.freescale.net/livelink/livelink?func=ll&objId=224108405&objAction=browse","'Files modified: Pwm.c Pwm_lpw.c Pwm_PBcfg.c

Table continues on the next page...

ID	Subtype	Headline and Description
		Pwm_Cfg.c Pwm_Cfg.h
ENGR00378994	NewWork	[PWM] Improve configuration structure layout for HW dependent parameters 'NewWork Description: After implementation of CR engr378611, the HW dependent configuration parameters were moved to IPW while this allowed a better separation for logical config params from HW-dependent config params it lead to a decrease in overall performance. Also, after discussions with tech-leads we concluded that the HW-dependent parameters (like Period, Idle-state, Polarity, default duty) should be stored in the HW-specific configuration structure. Requirement source: continuous improvement process Proposed solution (Optional): See analysis.","'Pwm_Cfg.c Pwm_PBcfg.c Pwm_Ipw.c
ENGR00371815	NewWork	[PWM] Improve placement of application-layer parameters in the configuration structure 'NewWork Description: Parameters like: Polarity, IdleState, default duty and period are high-layer parameters that have a strong dependence to the used HW timer. Due to separation of layer this parameters should not be placed in HW specific configuration part but in the IPWrapper part since this part make the link between High-Level and the HW layer. Requirement source: SW layered architecture concept for SMCAL Proposed solution (Optional): 1) Move the following parameters from the Pwm_[IPV]_ChannelConfigType (defined in Pwm_[IPV]_Types.h file) to Pwm_Ipw_Types.h: ePwmPolarity, ePwmIdleState, u16PwmDefaultDutyCycle, nPwmDefaultPeriod 2) Update Pwm_PBCfg.c and Pwm_Cfg.c file so that the correct structure is generated. Note. Extreme care needs to be taken for FlexPwm IPV. FlexPwm has one period defined for each submodule. For FlexPwm the following changes are needed. a) removed PwmDefaultPeriod from the configuration of each submodule (XDM) and enable it for editing at logical channel level b) add a check in the Pwm_[PB]Cfg.c files to verify if the same Period value was defined for all channels belonging to a certain submodule. If different value are written then user should receive an error. c) modify Pwm_FlexPwm_InitSubModuleRegs() by adding an new parameter (u16SubModPeriodValue) LOCAL_INLINE FUNC(void, PWM_CODE) Pwm_FlexPwm_InitSubModuleRegs /* @violates @ref Pwm_FlexPwm_C_REF_6 Identifier exceeds 31 chars. */ (P2CONST(Pwm_FlexPwm_SubModuleConfigType, AUTOMATIC, PWM_APPL_CONST) pFlexPwmSubModuleConfig, VAR (Pwm_FlexPwm_SubModuleType, AUTOMATIC) u8SubModuleId, VAR (Pwm_FlexPwm_SubModuleType, AUTOMATIC) u8UniqueSubModuleId,

Table continues on the next page...

ID	Subtype	Headline and Description
		VAR (uint16, AUTOMATIC) u16SubModPeriodValue, VAR (uint32, AUTOMATIC) u32ModRegAddr, VAR (uint32, AUTOMATIC) u32SubModRegAddr); And update /* Program period */ Pwm_FlexPwm_u16aRegPeriod[u8UniqueSubModuleId] = pFlexPwmSubModuleConfig->nPwmDefaultPeriod; to /* Program period */ Pwm_FlexPwm_u16aRegPeriod[u8UniqueSubModuleId] = u16SubModPeriodValue; d) in Pwm_FlexPwm_Init() change the initialization for all channels as following - If submodule was not initialized already Get configured period for a channel belonging to a submodule (which should be the same for all channels of one submodule -- see b)) call Pwm_FlexPwm_InitSubModuleRegs() with the given period value 3) Update Init function from Pwm_Ipw.c so that the to the lower levels also receives the values of the parameters given above 4) Update the IPV code to match the changed configuration types.","'Changed configuration structure so that HLD configuration data should not be configured at IP level. Affected files: Pwm_Ipw.c, Pwm_Ipw_Types.h, Pwm_Cfg.c, Pwm_Cfg_42.c, Pwm_PBCfg.c, Pwm_PBCfg_42.c.",ENGR00377839
ENGR00374183	NewWork	[PWM] Improvement for SRAM usage for configured channels. 'Initial Description: Currently the PWM driver is designed for performance optimization and simplification. However, there are some variables waste if only few PWM channels are configured. Pwm_GtmAtom_aChannelNotifType[PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmAtom_aChannelPeriod[PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmTom_aChannelPeriod[PWM_GTM_TOM_HW_CHANNELS_NO] PWM_GTM_ATOM_HW_CHANNELS_NO,PWM_GTM_TOM_HW_CHANNELS_NO are maximum channels supported in the device. However the customer may use only serveral channels (even only 1 channel). NOTE: Since similar approach was used for all other platforms a similar fix will be needed. Can PWM driver optimized to recude the SRAM waste? e.g Generate the above macro base on the number of configured channels. Solution: there are 3 arrays should be optimized: Pwm_GtmAtom_u16aChannelDuty [PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmAtom_aChannelPeriod [PWM_GTM_ATOM_HW_CHANNELS_NO] Pwm_GtmAtom_aChannelNotifType [PWM_GTM_ATOM_HW_CHANNELS_NO] All of them use pGtmAtomChannelConfig->nHwChannel as the index of arrays. This parameter contain the values of module ID and channel ID which has been encoded. Thus distribution of the elements were not continuous, it lead to producing the SRAM waste. On the other hand, above method has an advantage for Pwm_GtmAtom_aChannelNotifType. The processcommoninterrupt function using this array to enable higher layer interrupt handler when it is needed.

Table continues on the next page...

ID	Subtype	Headline and Description
		Since it is called from hardware, parameter is passed to it is only moduleID and channelID. Thus elements in this array can be get immediately. ISR need to process very fast so the calculating speed should be precedence over the memory size. Top 2 arrays (containing period and duty) should be optimized by following way: 1. Index the array using the order like CONST(Pwm_Gtm_ChannelConfigType, PWM_CONST) (*pChannelsConfig)[] . So distribution of the elements will be really continuous. Some functions, ex. Setdutycycle, GetOuputState,.. have parameter u8IdxChannelConfig, it can be used to get elements from array immediately without accessing to Pwm_Gtm_ChannelConfigType struct and get nHwChannel value. 2. The size of arrays is the number of channels configured. In the case configuring in PostBuild, the size of arrays is the maximum number perform by looping all configsets. The last array Pwm_GtmAtom_aChannelNotifType [] keep current indexing. The new type was created, it is Pwm_Gtm_EdgeNotificationType, allocate 1 byte in RAM, reduce 75%, less than old enum types Pwm_EdgeNotificationType",'Pwm_Cfg.h,ENGR00378208
ENGR00376913	Defect	[PWM] Incorrect range value of PwmChannelId 'Problem detailed description (how to reproduce it): Current range value of PwmChannelId in Pwm.xdm is 0->23 But Rayleigh only has 1 FTM module and 8 channels, so the range should be 0->7 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: range value of PwmChannelId in Pwm.xdm is 0->23 Expected behavior: the range should be 0->7 Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA","'Files modified: Pwm.xdm Pwm_Cfg.c Pwm_PBcfg.c
ENGR00369079	Defect	[PWM] Incorrect validation of notification parameters 'Problem detailed description (how to reproduce it): - PwmNotificationSupported = TRUE - For all Pwm channels: PwmNotification = NULL or NULL_PTR Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Observed behavior: Driver generates an error Expected behavior: There are not any errors or warnings generated Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove error generation in template file","'- Pwm_Cfg.h - Pwm_Cfg_42.h
ENGR00363127	Defect	[PWM] PWM_FEATURE_OPWFM switch ON when OPWMFB used 'Problem detailed description (how to reproduce it): When least 1 channel is configured OPWFMB mode, PWM_FEATURE_OPWFM is generated with value STD_ON, even nothing OPWFM channel is used. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: When least 1 channel is configured OPWFMB mode, PWM_FEATURE_OPWFM is generated with value STD_ON, even nothing OPWFM channel is used. Expected behavior: PWM_FEATURE_OPWFM do not be generated when OPWFMB channels be configrued 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 Pwm_Cfg.h [!ELSEIF "contains(node:ref(PwmeMiosChannel)/PwmModeSelect,'OPWFM')"!] [!VAR "opwfmfeature" = "1"!] should be changed: [!ELSEIF "(contains(node:ref(PwmeMiosChannel)/PwmModeSelect,'OPWFM')) and (not((contains(node:ref(PwmeMiosChannel)/ PwmModeSelect,'OPWFMB'))))"!] [!VAR "opwfmfeature" = "1"!]','Pwm_cfg.h
ENGR00380139	Defect	[PWM] Remove OPWMC feature. 'Problem detailed description (how to reproduce it): OPWMC updates duty immediately but not buffered. Typically, we have 2 A1 matches in a period, but in some cases, updating duty can generate 3 match in 1 period. The 3rd match causes the inverting of match order. The first match will be second match and the second match will be first match. We have output with inverting polarity. Duty cycle cannot be changed in run time, so this mode should be removed Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: NA Observed behavior: Sometime, changing duty via Pwm_SetDutyCycle cause inverting of polarity Expected behavior: Pwm_SetDutyCycle works correctly Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove OPWMC feature. Note that MPC574XG does not support OPWMC itself, just remove macro: #define PWM_FEATURE_OPWMC (STD_OFF)","'Pwm_eMios.c Pwm_eMios_Types.h",ENGR00380211
ENGR00374971	Defect	 [PWM] The edge enumerate table is incorrect 'Problem detailed description (how to reproduce it): The table 3-141 does not contain information about the initialization of the first element correctly. Name Initializer Description PWM_RISING_EDGE 0 A notification will be generated on the rising edge. According to the source , it should be 1. typedef enum { /** @brief A notification will be generated on the rising edge */ PWM_RISING_EDGE = 1, /** @brief A notification will be generated on the falling edge */ PWM_FALLING_EDGE, /** @brief A notification will be generated on any state transition */ PWM_BOTH_EDGES } Pwm_EdgeNotificationType; Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: The table 3-141 does not contain information about the initialization of the first element correctly. Expected behavior: The table 3-141 need to contain information about the initialization of the first element correctly. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'User Manual
ENGR00371917	Defect	 [PWM] Update for traceability improvement 'Problem detailed description (how to reproduce it): UML reviewing indicate that Pwm_eMios_SetPowerState should not have a "@implements" directive. Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Pwm_eMios_SetPowerState has a "@implements" directive. Expected behavior: Pwm_eMios_SetPowerState should not have a "@implements" directive. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove "@implements" directive above the function Pwm_eMios_SetPowerState",'Pwm_eMios.c
ENGR00373404	NewWork	[PWM] Use "READONLY" instead of "EDITABLE" in xdm file 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: N/A.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): n/a","'File modified: - Pwm.xdm
ENGR00364240	NewWork	[PWM] Version checking is missing in configuration template files 'NewWork Description: 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:

Table continues on the next page...

ID	Subtype	Headline and Description
		[!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_SP EC_VERSION_PATCH"!][!// [!VAR "ADC_SW_MAJOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MAJO R"!][!// [!VAR "ADC_SW_MINOR_VERSION_TEMPLATE"="M4_SRC_SW_VERSION_MINO R"!][!// [!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!][!// BLN_SMCAL_4.0_PANTHER_RTM_1.0.0 Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Please see Flexray, Adc driver, they are using *_VersionCheck.m. after that include this macro into template files","'Pwm_Cfg.h Pwm_Cfg.c Pwm_PBCfg.c Pwm_VersionCheck_Inc.m Pwm_VersionCheck_Src_PB.m Pwm_VersionCheck_Src.m
ENGR00364140	Defect	[PWM] parameter PwmPeriodInTicks cannot set FALSE 'Problem detailed description (how to reproduce it): PwmPeriodInTicks is a parameter in PwmChannel. When it is set FALSE (untick in the box), the value of period default is defined in parameter PwmPeriodDefault with unit Second. But generating alway returns error with any value of PwmPeriodDefault. This issue appears with channels which is config using below modes: - OPWMB - OPWM - OPWMCB - OPWMC Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: NA Observed behavior: parameter PwmPeriodInTicks cannot set FALSE Expected behavior: parameter PwmPeriodInTicks can set 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","'Pwm_Cfg.c Pwm_PBCfg.c
ENGR00363754	Defect	[PWM]DAOC mode can be used without notification supported 'Problem detailed description (how to reproduce it): DAOC mode use ISR to recalculate and update A,B register. A channel configure oparation mode as DAOC, it's flag enable bit must be set, user's interrupt handling also be called. If user want to use DAOC mode, he must configure PwmNotificationSupported parameter ENABLE and create a notification handling function. Expected behavior: DAOC mode can be used even PwmNotificationSupported parameter is DISABLE Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: DAOC always require notification support Expected behavior: DAOC does not require notification support Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Pwm_Cfg_42.h,ENGR00363763
ENGR00366559	Defect	[PWM]Pwm_eMios_SetPowerState mistake with Center Aligned modes 'Problem detailed description (how to reproduce it): OPWMCB and OPWMC mode use both internal bus masterbus. In current implementation, the function Pwm_eMios_SetPowerState check if a channel is configured with a mode which do not use internal counter, of course this channel will use masterbus, then driver will switch on or off clock which provides to the counter of masterbus depending power state. But the center aligned modes also use internal counter and it is not affected. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior:

Table continues on the next page...

ID	Subtype	Headline and Description
		Internal counter of center aligned mode is still running Expected behavior: Internal counter of center aligned mode stop Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In loop logic channel, add checking if channel is configured with CenterAligned-Mode, then process "power state" with it.",'Pwm_eMios.c
ENGR00369450	Defect	[SPI] Exception is generated when SPI driver was not initialized and an 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: pDspiDev 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 ENGR00369596 Requirement source: Design/Dev (SW) (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): N/A",'Spi_Cfg.h
ENGR00378000	Defect	[SPI] All status bits of DSPI_SR shall be cleared by deinit function 'Problem detailed description (how to reproduce it): All status bits of DSPI_SR shall be cleared by deinit function but currently only the status flags which are used by the driver in SPI mode are cleared. How to impact to functionality: - no impact Violated autosar requirement: SPI252 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: In Spi_Dspi_DeInit() function: REG_WRITE32(DSPI_SR_ADDR32(u8HWoffset), ((DSPI_SR_EOQF_W1C | DSPI_SR_RFDF_W1C) & DSPI_SR_RESERVED_MASK_U32)); Expected behavior: According to SPI252: In case of the SPI Handler/Driver state is not SPI_BUSY, the deInitialization function shall put all already initialized microcontroller SPI peripherals into the same state such as Power On Reset. =>So, All status bits of DSPI_SR shall be cleared. Proposed solution (Optional): This should be: REG_WRITE32(DSPI_SR_ADDR32(u8HWoffset),REG_READ32(DSPI_SR_AD

Table continues on the next page...

ID	Subtype	Headline and Description
		DR32(u8HWoffset)) & DSPI_SR_RESERVED_MASK_U32);","'- Reg_eSys_DSPI.h - Spi_DSPI.c
ENGR00371999	Defect	[SPI] Correct Misra Error and Compiler Warning 'Problem detailed description (how to reproduce it): Some misra error and compiler warning appear when testing for Racerunner IS RTM2.0.0 - In Spi_IPW.h file has some defines violated MISRA Rule 19.4: #define Spi_Ipw_apOneJobSeqsOptimized Spi_Dspi_apOneJobSeqsOptimized #define SPI_IPW_SET_CLOCK_MODE DSPI_SET_CLOCK_MODE - In Spi_Lcfg.c, Spi_Cfg.c, Spi_PBcfg.c files have violated MISRA Rule 10.1: In SpiChannelConfig_LT, SpiChannelConfig_PC, SpiChannelConfig_PB variables, Value of VAR(Spi_DataType, SPI_VAR) DefaultTransmitValue member is out of range Spi_DataType. - In Spi.c file: Variable "pcJobConfig" of Spi_SyncTransmit() function was set but never used. Preconditions: Check misra and compiler warning for some test case Spi_TS_Eq_Cot_01 ,Spi_TS_006, Spi_TS_032, Spi_TS_Define_Compile Test Case ID (internal TC that caught the defect) - optional Spi_TS_Eq_Cot_01 Spi_TS_006 Spi_TS_032 Spi_TS_Define_Compile Trigger: NA Observed behavior: Misra 19.4, 10.1 and compiler warning appear. Expected behavior: No error appear in Misra and compiler warning report Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): 1. Missra Error - In Spi_IPW.h: add comment for MISRA Rule Violated 19.4. - In Spi_Lcfg.c, Spi_Cfg.c, Spi_PBcfg.c: Change from: [!"SpiDefaultData"!]U, To: (Spi_DataType)[!"SpiDefaultData"!]U, 2. Compiler Warning: - In Spi_SyncTransmit() function of Spi.c: The "pcJobConfig" variable is need publish in SPI_DEV_ERROR_DETECT == STD_ON and SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON.","'Spi_IPW.h Spi_Lcfg.c, Spi_Cfg.c, Spi_PBcfg.c Spi.c
ENGR00373640	Defect	[SPI] Driver cannot discard last frame data in RX FIFO when the cancel of the sequence for Slave 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		In Slave mode: In case of setup the data length of Slave greater than the data length of Master. And use Spi_Cancel() function to cancel of the sequence for Slave after the sequence of Master is transfer complete. Driver cannot discard last frame data in RX FIFO when the cancel of the sequence for Slave. Because have one frame in the shift register is not put to RX FIFO if a transfer is in progress. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: Driver will discard all frame data in the RX FIFO when the cancel of the sequence for Slave. Proposed solution (Optional): We need use loop check state of module. Wait until state of module is Stopped before clear the TX FIFO and RX FIFO. Because have one frame in the shift register is not put to RX FIFO if a transfer is in progress.",'Spi_DSPI.c
ENGR00373130	Defect	[SPI] Error will appear when the SPI driver is uninitialized and a SPI interrupt appears 'Problem detailed description (how to reproduce it): In function Spi_Dspi_IsrFifoRx(), if the driver is uninitialized and an interrupt occurs, program will write to a wrong address, and system will crash. The section in which the flags must be cleared, at the end of the function, has a FOR statement that loops the HUnits. In order to do this, the driver needs the adreses, but if the driver is not initialized, these addresses are not known. Preconditions: Driver uninitialize and interrupt occur Test Case ID (internal TC that caught the defect) - optional Expected behavior: Driver will work normally","'Spi_DSPI.h, Spi_DSPI.c, Spi_Dspi_Irq.c
ENGR00369596	Defect	[SPI] Exception is generated when SPI driver was not initialized and an DSPI 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: pDspiDev 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: we run the test case with condition as: driver not initialized and the interrupt occurred

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional tsi_bbx_flt_spi_10101 Expected behavior: The driver can run normally and the interrupt flag will be cleared and test case is passed","'- Spi_DSPI.c - Reg_eSys_DSPI.h
ENGR00378336	NewWork	[SPI] 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 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: 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

Table continues on the next page...

ID	Subtype	Headline and Description
		update it according to the design Output: - 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: Planned requirement Proposed solution (Optional): Fill in the attached template Exclusive areas defined in driver code Updated IM chapter 5.1 according to exclusive areas defined in the code",'Add in driver/doc/design/ the report AUTOSAR_MCAL_SPI_EXCLUSIVE_AREAS.xls(Please see in file attached to CR).
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. Requirement source: Customer request","'-spi\generic\doc\docgen\um-xml\topics\sub_driver \sub_doxygen\sub_defines -runtime_errors.xml
ENGR00363317	Defect	[SPI] Inconsistent handling of Spi memory sections 'Problem detailed description (how to reproduce it): The handling of Spi memory sections is inconsistent: In Spi_DSPI.h, line 280, variable section SPI_START_SEC_VAR_NO_INIT_UNSPECIFIED is opened, but not closed until line 367. Meanwhile, in line 312, a new section is opened and that leads to a compiler error. Preconditions: Any. Observed behavior: In Spi_DSPI.h, line 280, variable section SPI_START_SEC_VAR_NO_INIT_UNSPECIFIED is opened, but not closed until line 367. Expected behavior: Correct the memory definition section. Proposed solution (Optional): Correct the memory definition section : SPI_START_SEC_VAR_NO_INIT_UNSPECIFIED",'- Update the definition memory section SPI_START_SEC_VAR_NO_INIT_UNSPECIFIED in Spi_DSPI.h
ENGR00379929	Defect	[SPI] Incorrect description of SpiPhyUnitSync 'Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		The SpiPhyUnitSync has following description under EB Tresos: Specific if this HwUnit can only do sync transfers. If true then this hardware unit is dedicated for Synchronous transfers only. If false then this hardware unit is dedicated for Asynchronous transfers only. False is applicable only if SpiGeneral/SpiLevelDelivered is either 1 or 2 and true is applicable only if SpiGeneral/SpiLevelDelivered is 0. It should be: Specific if this HwUnit can only do sync transfers. If true then this hardware unit is dedicated for Synchronous transfers only. If false then this hardware unit is dedicated for Asynchronous transfers only. False is applicable only if SpiGeneral/SpiLevelDelivered is either 1 or 2 and true is applicable only if SpiGeneral/SpiLevelDelivered is 0 or 2. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Incorrect description Expected behavior: Correct 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","'-Spi.xdm -SpiPhyUnit.xml
ENGR00365672	Defect	[SPI] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): 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 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 Observed behavior: Incorrect version checking in configuration template files: - [DRV]_VersionCheck_Inc.m - [DRV]_VersionCheck_Src.m - [DRV]_VersionCheck_Src_PB.m Expected behavior: Correct version checking in configuration template files. Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		1. Replace: - ArMajorVersion with ArReleaseMajorVersion - ArMinorVersion with ArReleaseMinorVersion - ArPatchVersion with ArReleaseRevisionVersion 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","'Changed files: - Spi_VersionCheck_Inc.m - Spi_VersionCheck_Src.m - Spi_VersionCheck_Src_LT.m - Spi_VersionCheck_Src_PB.m
ENGR00373896	NewWork	[SPI] Investigate and implement to reduce cyclomatic complexity and nesting level 'NewWork Description: In AUTOSAR_MCAL_<MDL>_StaticAnalysis_Summary.xlsx, at Function Details 2 new columns are added: - Cyclomatic Complexity Comments - Nesting Depth Comments - for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth >4 the problems shall be resolved Requirement source: Requirements: - 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. 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 Proposed solution (Optional): - for all the cyclomatic complexity <= 9 the value shall be N/A in the comment - for all the cyclomatic complexity between [10-20] the deviation reason shall be added in the comment column - for all the cyclomatic complexity >20 the problems shall be resolved - for all the nesting depth between [0 4] the value shall be N/A in the comment - for all the nesting depth >4 the problems shall be resolved","'-Spi_DSPI.c -Spi_DSPI.h
ENGR00365712	Defect	[SPI] Missing Compiler Switch

Table continues on the next page...

ID	Subtype	Headline and Description
		DISABLE_MCAL_INTERMODULE_ASR_CHECK 'Problem detailed description (how to reproduce it): In Spi_Cfg.c, line 551, the compiler switch DISABLE_MCAL_INTERMODULE_ASR_CHECK is missing Preconditions: Any Observed behavior: In Spi_Cfg.c, line 551, the compiler switch DISABLE_MCAL_INTERMODULE_ASR_CHECK is missing Expected behavior: Add the compiler switch DISABLE_MCAL_INTERMODULE_ASR_CHECK","'- Spi_Cfg.c - Spi_PBcfg.c - Spi_Lcfg.c
ENGR00366583	Defect	[SPI] Newly found errata in the SPI 'NewWork Description: Hello, Here are errata that is sent from customer: - On parity error scenario in master mode the frame transfer does not restart even if interrupt is properly served and flag is cleared. Found during the validation of a different SoC - When there is a receive FIFO overflow, data may be loaded into the receive FIFO even after the overflow flag is cleared. Found during the validation of a different SoC. Please analysis and provide the final decision => yes/no we could be implement these errata in MCAL product. - It relate to parity, perhap this errata will be not cover by MCAL product. It should be done by customer side, to verify data before transferring data by protocol. - Overflow sometimes that appear in spi slave, i think this errata could be done by mcal product. Requirement source: Customer Request",'Code must be updated to discard the data when overrun is detected
ENGR00372513	NewWork	[SPI] Rename SpiSyncTransmitTimeout node in xdm file 'NewWork Description: Define SPI_TIMEOUT_COUNTER have using for purpose SyncTransmit and AsyncTransmit. So, we need rename SpiSyncTransmitTimeout node in xdm file. Requirement source: Dev (SW) Proposed solution (Optional): Rename SpiSyncTransmitTimeout to SpiTransmitTimeout. Update comment for this node.","'Changed files: -Spi.xdm -Spi_Cfg.h
ENGR00372698	Defect	[SPI] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): Some plugin files contain hard-coded copyright Preconditions: No special preconditions Test Case ID (internal TC that caught the defect) - optional None Trigger: None Observed behavior: In some plugin files the copyright is hardcoded Expected behavior: In all driver files the copyright shall be M4_SRC_COPYRIGHTED_TO to be replaced when building the plugin with the correct information. Proposed solution (Optional): Replace hard-coded copyright to M4_SRC_COPYRIGHTED_TO",'In spi\specific \generate *.m files.
ENGR00364456	Defect	[SPI] SPI sequence can't finish when SPI_FORCE_DATA_TYPE == STD_ON 'Problem detailed description (how to reproduce it): With configuration SPI_FORCE_DATA_TYPE == STD_ON and DATA_FRAME = 16bit, the test will be not run, because the current sequence can't finish Test Case ID (internal TC that caught the defect) - optional Spi_TS_018 Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: NA Expected behavior: The test will be passed Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA",'Update the condition of checking end of job
ENGR00372425	Defect	[SPI] Slave cannot discard last byte in receive buffer when TX FIFO underflow has occurred 'Problem detailed description (how to reproduce it): In case of low baudrate(Eg: 1000bps), and Master transmits a sequence with a data length that is greater than Slave expects, Slave may assume that the extra byte are belong to a new sequence. In case of Slave use DMA to transfer: Slave cannot discard the receive data after finished job. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		NA Trigger: NA Observed behavior: NA Expected behavior: Slave will discard extra byte before preparing for a new sequence. Proposed solution (Optional): - Slave will clear TFUF flag in RFDF interrupt. - In Spi_Dspi_JobTransfer(): Slave will wait until TFUF flag is cleared before prepare next Job of new sequence.","'- Spi_DSPI.c - Reg_eSys_DSPI.h
ENGR00373375	NewWork	[SPI] Some Non-Autosar confiugration parameters should have OPTIONAL attribute 'NewWork Description: Summary the discussion between TL and CE: Parameters under certain condition - All reference nodes will be updated to be marked as optional - All parameters that can be marked as optional, but have already a default value, can be kept as mandatory with default value (using the READONLY to enable the node). Implementation details The parameters that can be excluded from a specific configuration shall be marked as optional (<a:a name="OPTIONAL" value="true"/>, which results in lower multiplicity =0 and upper multiplicity 1). The AUTOSAR parameters have already defined the lower and the upper multiplicity. This change is needed only for the NON-Autosar parameters. The boolean Non-Autosar parameters that enable / disable Non-Autosar features must not be marked as OPTIONAL - only the additional configuration parameters that are needed only when Non-Autosar features are enabled. All parameters that are under this category shall be marked as OPTIONAL=true in the xdm. The check that the features were correctly configured shall be transferred to the boolean parameter that enables the feature. Example: Node1 - configuration node for Non-Autosar feature that selects between transfer mode interrupt or DMA should not be OPTIONAL Node2 - configuration node that selects Dma channel reference needed when DMA transfer is selected should be OPTIONAL when Node1 selects DMA, there should be added an invalid check for the existence of Node2 Reference point SpiPhyTxDmaChannel will be enabled when SpiPhyUnitAsyncMethod is set up to be DMA the changes: into <v:var name="SpiPhyUnitAsyncMethod" type="ENUMERATION"> [...to be added the check that everything is configured the INVALID tag] <a:da name="INVALID" type="XPath"> <a:tst expr="((node:value(.)='DMA') and (node:exists(../SpiPhyTxDmaChannel))) or not(node:value(.)='DMA')" false="Please enable SpiPhyTxDmaChannel"/> </a:da> <v:ref name="SpiPhyTxDmaChannel" type="CHOICE-REFERENCE"> [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		<a:a name="OPTIONAL" value="true"/> <a:da name="EDITABLE" type="XPath" expr="not(node:value(../ SpiPhyUnitAsyncMethod) = 'DMA')"/> ---------- Original request from customer engineer---------- Parameters: - Non-autosar parameters. - Specific parameters under certain conditions. This is my additional details: Customer don't use some reference nodes (DMA with Spi, Clock reference with GPT...). Epc file that exported from EB tresos with no reference value in DMA SPI case. For ex: <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/ TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyRxDmaChannel</ DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/ TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannel</ DEFINITION-REF> </ECUC-REFERENCE-VALUE> <ECUC-REFERENCE-VALUE> <DEFINITION-REF DEST="ECUC-CHOICE-REFERENCE-DEF">/ TS_T2D47M8I0R0/Spi/SpiGeneral/SpiPhyUnit/SpiPhyTxDmaChannelAux</ DEFINITION-REF> </ECUC-REFERENCE-VALUE> But in EPD file, these parameters are defined as mandatory with lower/upper multiplicity =1. So they should be have default value because a mandatory parameter cannot be empty. Requirement source: Customer request Proposed solution: 1. The not used parameters _should_not_ be part of the Tresos Export. So please change the xdm-file in that way that the parameters will not be part of the export for the case that they are not used! 2. Lower multiplicity =0 Upper multiplicity =1 for optional. ----------","'Files change: -Spi.xdm -Spi_Cfg.h -Spi_Cfg.c -Spi_PBcfg.c -Spi_Lcfg.c -some file in test\test_spi folder.
ENGR00344789	Defect	 [SPI] Some local variables are wrongly guarded with SPI_DEV_ERROR_DETECT 'Problem detailed description (how to reproduce it): Some local variables in Spi_SyncTransmit() are guarded by SPI_DEV_ERROR_DETECT=STD_ON: P2CONST(Spi_SequenceConfigType, AUTOMATIC, SPI_APPL_CONST) pcSequenceConfig;

Table continues on the next page...

ID	Subtype	Headline and Description
		VAR(Spi_JobType, AUTOMATIC) Job; VAR(Spi_JobType, AUTOMATIC) JobIndex; VAR(Spi_JobType, AUTOMATIC) NumJobsInSequence; P2CONST(Spi_JobConfigType, AUTOMATIC, SPI_APPL_CONST) pcJobConfig; but they are used also when DET is OFF. Also, only the checks and calls to Det_ReportError() function should be guarded by SPI_DEV_ERROR_DETECT=STD_ON. Preconditions: SPI_DEV_ERROR_DETECT=STD_ON Trigger: Call Spi_SyncTransmit(). Observed behavior: When DET is OFF, an error occur saying that some variables are not declared. Expected behavior: - Some local variables must not be guarded by SPI_DEV_ERROR_DETECT=STD_ON: P2CONST(Spi_SequenceConfigType, AUTOMATIC, SPI_APPL_CONST) pcSequenceConfig; VAR(Spi_JobType, AUTOMATIC) Job; VAR(Spi_JobType, AUTOMATIC) JobIndex; VAR(Spi_JobType, AUTOMATIC) NumJobsInSequence; P2CONST(Spi_JobConfigType, AUTOMATIC, SPI_APPL_CONST) pcJobConfig; - Det_ReportError() function should be guarded by SPI_DEV_ERROR_DETECT=STD_ON. Proposed solution (Optional): Those local variables must not be guarded by SPI_DEV_ERROR_DETECT=STD_ON Only the checks in which Det_ReportError() function is called should be guarded by SPI_DEV_ERROR_DETECT=STD_ON.",'-Spi.c
ENGR00369472	Defect	[SPI] Start and End job notifications behavior while used GPIO pins to control chip selection 'Problem detailed description (how to reproduce it): We are now getting to the point where we're using the Start and End notifications to demux our CS signals for various SPI devices. However, we've noticed that when there are multiple Jobs within a Sequence, StartNotifications for each job within the sequence is executed at the same time in loop, before the first actual Job starts. This means that by the time the first Job starts, the StartNotification for the last Job has executed already and in our case has already set the CS for the last job. I understand that StartNotifications are not a part of the AUTOSAR specifications. However, based on prior emails with Freescale the StartNotification has been added in order to support this type of scenario. My expectation would be that for any given sequence it would follow something like this: Sequence 1: Job1 -> StartNotification -> Execute Job 1 -> EndNotification -> Job2 -> StartNotification -> Execute Job 2 -> EndNotification Right now we are seeing something like this: Sequence 1: StartNotification(Job1) -> StartNotification(Job2) -> Execute Job1 -> End Notification(Job1) -> Execute Job2 -> EndNotification(Job2) CE suggestion:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Customer would like to use GPIO pins to control chip selection instead of CS pins of SPI. so they will active SPI slave via notification. - If the notification is call that mean job should be transmit immediately then. Preconditions: Use GPIO pins to control chip selection instead of CS pins of SPI. Observed behavior: Sequence 1: StartNotification(Job1) -> StartNotification(Job2) -> Execute Job1 -> End Notification(Job1) -> Execute Job2 -> EndNotification(Job2) Expected behavior: Sequence 1: Job1 -> StartNotification -> Execute Job 1 -> EndNotification -> Job2 -> StartNotification -> Execute Job 2 -> EndNotification - If the notification is call that mean job should be transmit immediately then. Proposed solution (Optional): remove job notificaion in function Spi_AsyncTransmit() and put them into Spi_LLD_JobTransfer()",'- Spi.c
ENGR00370354	NewWork	[SPI] Start and End job notifications behavior while used GPIO pins to control chip selection 'NewWork Description: We are now getting to the point where we're using the Start and End notifications to demux our CS signals for various SPI devices. However, we've noticed that when there are multiple Jobs within a Sequence, StartNotifications for each job within the sequence is executed at the same time in loop, before the first actual Job starts. This means that by the time the first Job starts, the StartNotification for the last Job has executed already and in our case has already set the CS for the last job. I understand that StartNotifications are not a part of the AUTOSAR specifications. However, based on prior emails with Freescale the StartNotification has been added in order to support this type of scenario. My expectation would be that for any given sequence it would follow something like this: Sequence 1: Job1 -> StartNotification -> Execute Job 1 -> EndNotification -> Job2 -> StartNotification -> Execute Job 2 -> EndNotification Right now we are seeing something like this: Sequence 1: StartNotification(Job1) -> StartNotification(Job2) -> Execute Job1 -> End Notification(Job1) -> Execute Job2 -> EndNotification(Job2) CE suggestion: - Customer would like to use GPIO pins to control chip selection instead of CS pins of SPI. so they will active SPI slave via notification. - If the notification is call that mean job should be transmit immediately then. Requirement source: Customer request Proposed solution (Optional): remove job notificaion in function Spi_AsyncTransmit() and put them into Spi_LLD_JobTransfer()",'- Spi_DSPI.c
ENGR00369598	Defect	[SPI] The slave will lead to an exception if the master send extra frames in DMA mode

Table continues on the next page...

ID	Subtype	Headline and Description
		'Problem detailed description (how to reproduce it): In Slave use DMA mode: The number of frame in RX FIFO is not discard when the master send extra frames. Preconditions: In Slave use DMA mode: Setup TX data size on Master device is more greater than expected RX data size on Slave device Test Case ID (internal TC that caught the defect) - optional Spi_TS_023 Observed behavior: The number of frame in RX FIFO is not discard. Test suite Spi_TS_023 is false. Expected behavior: Test case is passed","'Changed file: - Spi_DSPI.c
ENGR00371600	Defect	[SPI] Unable to cancel SPI sequence when external SPI master stop without finished job trasmission 'Problem detailed description (how to reproduce it): we are using your MCAL SPI driver for the MPC5744P. We have several SPI master connections running and one SPI slave. All units do use the Async-interface. We encountered the following problem: When the external SPI master which sends to our slave port does stop an ongoing transmission without finishing the transmission completely (i.e. not all bytes are sent), the call of Spi_Cancel(Slave_sequence) does not produce the desired result, that is, our slave interface never terminates the ongoing job and therefore never reaches the cancelled state. Further receiving on the slave interface is impossible after such a break. Please note that it is not possible to reset the interface with Spi_Init() as this would disturb the other SPI connections. Can you describe a solution to the problem of cancelling a slave sequence? Preconditions: The SPI driver is in Slave mode. when external SPI master stop without finished job trasmission. Observed behavior: Unable to cancel SPI sequence for the Slave. Expected behavior: In Spi_Cancel(): - If sequence that is Slave -> stop sequence immediately and transfer another sequence while query not emty . Spi slave will not depend on external device (master). - In Master case should work as normally.","'Changed file: - Spi.c - Spi_IPW.h",ENGR00373001
ENGR00370121	NewWork	[SPI] Update Base address array to display all HW Units 'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = {

Table continues on the next page...

ID	Subtype	Headline and Description
		#ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the address 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....","'Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the address 0x0 will be generated:
ENGR00365645	NewWork	[SPI] Update SequenceCacheAttrsType and SET_CLOCK_MODE to serve multiple IPVs 'NewWork Description: Under the define "SPI_OPTIMIZE_ONE_JOB_SEQUENCES" , a typedef maned Spi_Dspi_SequenceCacheAttrsType and DSPI_SET_CLOCK_MODE exists. But on KinetisKFA there is Spi_LPspi_SequenceCacheAttrsType, LPSPI_SET_CLOCK_MODE So need to find a method to make generic this type. Proposed solution (Optional): - In Spi.c rename: +"Spi_Dspi_SequenceCacheAttrsType" to "Spi_HW_SequenceCacheAttrsType". +"Spi_Dspi_apOneJobSeqsOptimized" to "Spi_Ipw_apOneJobSeqsOptimized". +"DSPI_SET_CLOCK_MODE" to "SPI_IPW_SET_CLOCK_MODE" - In Spi_IPW.h put: #if (SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON)

Table continues on the next page...

ID	Subtype	Headline and Description
		typedef Spi_LPspi_SequenceCacheAttrsType Spi_HW_SequenceCacheAttrsType; #define Spi_Ipw_apOneJobSeqsOptimized Spi_LPspi_apOneJobSeqsOptimized #endif /*SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON*/ #if (SPI_DUAL_CLOCK_MODE == STD_ON) #define SPI_IPW_SET_CLOCK_MODE LPSPI_SET_CLOCK_MODE #endif /*SPI_DUAL_CLOCK_MODE == STD_ON*/ - And on all the other platforms, in Spi_IPW.h: #if (SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON) typedef Spi_Dspi_SequenceCacheAttrsType Spi_HW_SequenceCacheAttrsType; #define Spi_Ipw_apOneJobSeqsOptimized Spi_Dspi_apOneJobSeqsOptimized #endif /*SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON*/ #if (SPI_DUAL_CLOCK_MODE == STD_ON) #define SPI_IPW_SET_CLOCK_MODE DSPI_SET_CLOCK_MODE #endif /*SPI_DUAL_CLOCK_MODE == STD_ON*/","'Spi.c Spi_IPW.h
ENGR00379422	NewWork	[SPI] Update UM with description of timeout functionality and recommendations for configuring timeout parameters 'NewWork Description: -Update UM with description and recommendations for configuring timeout parameters. Expected behavior: -Update the description for timeout value field in the plugin to include a recommended value Proposed solution (Optional): -Add a recommended value for the timeout value from the plugin.","'Update UM with description and recommendations for configuring timeout parameters. The Chapters need to update: - Chapter 3.7.1.50 Define SPI_TIMEOUT_COUNTER - Chapter 3.7.1.54 Define SPI_WAIT_LOOP_TICKS - Chapter 4.4.10 SpiSyncTransmitTimeout (SpiGeneral) - Chapter 4.4.14 SpiClockRef (SpiGeneral) * Files/dirs affected: -drivers\AutoSAR\spi\doc\user_manuals\docgen\um-xml\images\tresos -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\images\tresos -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\images\tresos \TimePerFrame_SpiCsContinous_False.png -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\images\tresos \TimePerFrame_SpiCsContinous_True.png -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\topics\sub_driver \sub_doxygen\sub_defines\SPI_TIMEOUT_COUNTER.xml -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\topics\sub_driver \sub_doxygen\sub_defines\SPI_WAIT_LOOP_TICKS.xml -drivers\AutoSAR\spi\generic\doc\docgen\um-xml\topics\sub_tresos \SpiGeneral.xml
ENGR00377905	Defect	[SPI] Update code generation when DMA is enabled 'Problem detailed description (how to reproduce it): When SpiLevelDelivered = 1 or 2(SPI_LEVEL_DELIVERED = LEVEL1 or

Table continues on the next page...

ID	Subtype	Headline and Description
		SPI_LEVEL_DELIVERED = LEVEL2) and SpiGlobalDmaEnable = true(SPI_DMA_USED = STD_ON). But the SPI unit is not configured to use DMA to transfer the data(SpiPhyUnitAsyncMethod=PIO_FIFO) or only use to Synchronous transfers(SpiPhyUnitSync = true). In the generated files(Spi_Cfg.c, Spi_Lcfg.c and Spi_PBcfg.c), some configuration structures (HWUnitConfig_PC, HWUnitConfig_LT, HWUnitConfig_PB) will not initialize values for some members (u16UseDma, u8TxDmaChannel, u8TxDmaChannelAux, u8RxDmaChannel). This makes the Configuration structures not to have correct values for those members. Preconditions: the defines (SPI_LEVEL_DELIVERED = LEVEL1 or SPI_LEVEL_DELIVERED = LEVEL2) and SPI_DMA_USED = STD_ON. the SPI unit do not configuration to use DMA to transfer the data(SpiPhyUnitAsyncMethod=PIO_FIFO) or only use to Synchronous transfers(SpiPhyUnitSync = true). Test Case ID (internal TC that caught the defect) - optional Spi_TS_003 Trigger: NA Observed behavior: In Spi_Cfg.c, Spi_Lcfg.c and Spi_PBcfg.c: Some variables(HWUnitConfig_PC, HWUnitConfig_LT, HWUnitConfig_PB) are missing elements(u16UseDma, u8TxDmaChannel, u8TxDmaChannelAux, u8RxDmaChannel). So, it do not make to build error because the code do not use those members but make a compile warning: missing initializer for those field. Expected behavior: Update code generation when DMA is enabled so the generated structures will contain also the missing elements(u16UseDma, u8TxDmaChannel, u8TxDmaChannelAux, u8RxDmaChannel) 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_Cfg.c, Spi_Lcfg.c and Spi_PBcfg.c: Replace: [!IF "SpiPhyUnitSync = 'true'"!][!// , SPI_PHYUNIT_SYNC_U32[!// [!ELSE!][!// , SPI_PHYUNIT_ASYNC_U32[!// [!IF "ecu:get('SpiDMAPresent') = 'TRUE' and (../../SpiGlobalDmaEnable = 'true')"!][!// [!IF "SpiPhyUnitAsyncMethod = 'DMA'"!][!// , (uint16)TRUE , [!"node:value(node:ref(SpiPhyTxDmaChannel)/MclDMAChannelId)"!]u , [!"node:value(node:ref(SpiPhyTxDmaChannelAux)/MclDMAChannelId)"!]u , [!"node:value(node:ref(SpiPhyRxDmaChannel)/MclDMAChannelId)"!]u [!ENDIF!][!// [!ENDIF!][!// [!ENDIF!][!// with: [!IF "SpiPhyUnitSync = 'true'"!][!// , SPI_PHYUNIT_SYNC_U32[!// [!IF "ecu:get('SpiDMAPresent') = 'TRUE' and ../../SpiGlobalDmaEnable = 'true' and ../../SpiLevelDelivered >0"!][!// , (uint16)FALSE , 0u

Table continues on the next page...

ID	Subtype	Headline and Description
		, 0u , 0u [!ENDIF!][!// [!ELSE!][!// , SPI_PHYUNIT_ASYNC_U32[!// [!IF "ecu:get('SpiDMAPresent') = 'TRUE' and ../../SpiGlobalDmaEnable = 'true' and ../../SpiLevelDelivered >0"!][!// [!IF "SpiPhyUnitAsyncMethod = 'DMA'"!][!// , (uint16)TRUE , [!"node:value(node:ref(SpiPhyTxDmaChannel)/MclDMAChannelId)"!]u , [!"node:value(node:ref(SpiPhyTxDmaChannelAux)/MclDMAChannelId)"!]u , [!"node:value(node:ref(SpiPhyRxDmaChannel)/MclDMAChannelId)"!]u [!ELSE!][!// , (uint16)FALSE , 0u , 0u , 0u [!ENDIF!][!// [!ENDIF!][!// [!ENDIF!][!//","'- Spi_Cfg.c - Spi_Lcfg.c - Spi_PBcfg.c
ENGR00348487	NewWork	[SPI] Update info about Slave baudrate 'NewWork Description: 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. Requirement source: Customer request Proposed solution (Optional): The SpiBaudrate tag, in xdm file, is used only for MASTER mode. For Slave mode, the baud rate is driven by the master. So in our case, we don't need this value (the registers for SLAVE do not contain information's about baudrate). Maybe a note in the description of this field, saying that this field is used only in MASTER mode, will help.","'The SpiBaudrate Range tag, in xdm file, must be updated. For Slave mode, the upper limit must be updated to 60 MHz.
ENGR00373406	NewWork	[SPI] Use "READONLY" instead of "EDITABLE" in xdm file 'NewWork Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters.

Table continues on the next page...

ID	Subtype	Headline and Description
		EDITABLE: Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly. Requirement source: Customer request",'- Spi.xdm
ENGR00379831	NewWork	[SPI] Use CPU clock to calculate value of the define SPI_TIMEOUT_COUNTER 'NewWork Description: - The SPI driver need use value reference to CPU Clock to calculate value of the define SPI_TIMEOUT_COUNTER. But currently, It is using the same value reference with SPI clock source configuration by SpiClockRef node. So, we need add reference node to CPU clock to use for calculate value of the define SPI_TIMEOUT_COUNTER. Requirement source: Use CPU clock to calculate value of the define SPI_TIMEOUT_COUNTER. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): - For the platforms(IMXVS4,MAC57DXXX,MAC58RXXX,MPC567XF_4.2,MPC574XP,MPC5775K,MPC577XN_ES,MPC5777C,S32R274_4.2): Add new reference node(SpiCPUClockRef) to CPU clock to use for calculate value of the define SPI_TIMEOUT_COUNTER. - For the platforms(MPC577XM,MPC574XG,MPC574XR,S32K14X,S32K14X_4.2): We will rename SpiClockRef node with SpiCPUClockRef and use to calculate value of the define SPI_TIMEOUT_COUNTER. Because they are using SpiPhyUnitClockRef node instead of SpiClockRef node for reference to the SPI clock source configuration.","'Files/dirs affected: -Spi.xdm -Spi_Cfg.h -spi\generic\doc\docgen\um-xml\topics\sub_driver\sub_doxygen\sub_defines\SPI_TIMEOUT_COUNTER.xml -spi\generic\doc\docgen\um-xml\topics\sub_driver\sub_doxygen\sub_defines\SPI_WAIT_LOOP_TICKS.xml -spi\generic\doc\docgen\um-xml\topics\sub_tresos\SpiGeneral.xml -dev\test\test_spi\specific\cfg\cfg_mcp\Spi.epc -dev\test\test_spi\specific\cfg\specific_shared\Mcu.epc
ENGR00378355	NewWork	[SPI] Verify all Misra errors and comments

Table continues on the next page...

ID	Subtype	Headline and Description
		'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA","'-Spi.h -Spi.c -Spi_Cfg.h -Spi_Cfg.c -Spi_Lcfg.c -Spi_PBcfg.c -Spi_IPW_Types.h",ENGR00379099
ENGR00364237	Defect	[SPI] 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"!][!// [!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

Table continues on the next page...

ID	Subtype	Headline and Description
		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","'- Spi_Cfg.c - Spi_PBcfg.c - Spi_Lcfg.c - Spi_Cfg.h - Spi_VersionCheck_Inc.m - Spi_VersionCheck_Src.m - Spi_VersionCheck_Src_PB.m - Add Spi_VersionCheck_Src_LT.m file - Spi.mak
ENGR00374730	Defect	 [WDGIF] Wdg_SetMode and Wdg_SetTriggerCondition cannot be reached via function pointer 'Problem detailed description (how to reproduce it): Issue Type: Bug Issue classification: low(C) Priority: urgent Package: Wdg S32K14X Found in version: MCAL4_0_BETA_0_9_0 What happens (symptoms): ---------- Vector WdgIf calls Wdg API function Wdg_SetMode via function pointer, but it cannot be reached, because the Wdg driver uses function like macro to redefine Wdg_SetMode to Wdg_ChannelSetMode: #define Wdg_SetMode(Mode) Wdg_ChannelSetMode(Mode, WDG_IPW_INSTANCE0) According BSW00330 in AUTOSAR_SWS_WatchdogDriver.pdf (Page47, 12: Not applicable requirements) the requirement for usage of macros / inline function instead of functions has been skipped and would not be allowed. When does this happen: ---------- When WdgIf tries to locate Wdg_SetMode() API. In which configuration does this happen: ---------- Any Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]","'wdgif means to be higher layer of wdg. A generic implementation will make use of pointers array of function API in WDG. incase of wdg have one instance, wdgif still use function-like macro instead of pointers array
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): Add constraints in Wdg.xdm configuration file for:WdgClockValue,WdgTimeoutPeriod,WdgWindowPeriod.",'Add constraints in configuration to validate the inputs
ENGR00371984	NewWork	[WDG] Add suport for ASR 4.2.1 to the RaceRunner Ultra platform 'NewWork Description: Integrate the ASR 4.2.1 implementation from the Cobra ASR 4.2.1 to the RaceRunner Ultra platform. You should integrate the changes from to the dev branches done for Cobra ASR 4.2.1 for WDG(see linked CR) Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Add suport for ASR 4.2.1 to the RaceRunner Ultra platform
ENGR00378830	Defect	[WDG] Compiler warning Swt_u32InitialKey is not initialized 'Problem detailed description (how to reproduce it): The customer reported following compiler warning with S32V234 MCAL0.8.1. generated\src\Wdg_43_Instance0_Cfg.c, 193: warning: missing initializer for field 'Swt_u32InitialKey' of 'Wdg_Swt_ConfigType' [-Wmissing-field-initializers] Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		Attached configuration (WdgKeyedService = false) Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Compiler warning as above Expected behavior: No compiler warning as above Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):","'Even OFF_MODE doesn't need data for initialkey but for C programing need to fill enough data for struct has type Wdg_Swt_ConfigType otherwise, it will have a compiler warning. Missing default value(-Wmissing-field-initializers) of Swt_u32InitialKey in offmode_setting. The Cr was raised to solve compiler warning.
ENGR00378343	Defect	[WDG] Exclusive area missing 'Problem detailed description (how to reproduce it): Missing exclusive area to protect the Wdg_aeStatus variable. Preconditions: In case Wdg_ChannelValidateGlobalCall and Wdg_ChannelEndValidateGlobalCall may interrupt each other. Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: Exclusive areas are missing. This might lead to inconsistencies. Expected behavior: Exclusive area must be used to protect critical sections. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Exclusive area must be used to protect variable Wdg_aeStatus.","'remove old implemetation introduce new one following the rules",ENGR00379177
ENGR00369518	NewWork	[WDG] Fix compiler warning issues by IAR compiler 'NewWork Description: investigate and fix compiler warnings. Expected behavior: no compiler warning Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): the root causes compiler warning because IAR compiler doesn't consider array

Table continues on the next page...

ID	Subtype	Headline and Description
		pointer Wdg_pConfigPC as a const possible solution: change P2CONST(Wdg_ConfigType, WDG_VAR, WDG_APPL_CONST) Wdg_pConfigPC[WDG_NO_OF_INSTANCES] to CONSTP2CONST(Wdg_ConfigType, WDG_VAR, WDG_APPL_CONST) Wdg_pConfigPC[WDG_NO_OF_INSTANCES]",'Fix compiler warning issues by IAR compiler
ENGR00381371	NewWork	[WDG] Fixes MISRA error(s) on Calypso RTM1.0.2 'NewWork Description: There are some misra issue concern rule 9.2 and rule 11.3 on Calypso RTM1.0.2. So, please fixes MISRA error(s) Requirement source: NA (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fixes MISRA error(s) on Calypso RTM1.0.2",'Fix misra error(s)
ENGR00368581	Defect	[WDG] 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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];",'Corrected the MemMap section allocation in the driver.,ENGR00368940
ENGR00373424	Defect	 [WDG] Inconsistent/Incorrect usage of Memory Allocation Sections (part 2) 'Problem detailed description (how to reproduce it): There is an In-consistence/Incorrect usage of Memory Sections in WDG driver: -ICU_APPL_CONST is incorrectly used in WDG (instead of WDG_APPL_CONST) -DEM error codes must be moved from SEC_CONFIG_DATA_UNSPECIFIED to SEC_CONST_UNSPECIFIED -Wdg_au8Index must be moved from WDG_START_SEC_CONFIG_DATA_8 to WDG_START_SEC_CONST_8 Observed behavior: Variables are mapped to incorrect sections. Expected behavior: All global variables used by Mcal drivers must be mapped to the correct memory sections. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): -Update form ICU_APPL_CONST to WDG_APPL_CONST -DEM error codes must be moved from SEC_CONFIG_DATA_UNSPECIFIED to SEC_CONST_UNSPECIFIED -Wdg_au8Index must be moved from WDG_START_SEC_CONFIG_DATA_8 to WDG_START_SEC_CONST_8",'Review and correct memory allocation
ENGR00365673	Defect	 [WDG] Incorrect version checking in configuration template files 'Problem detailed description (how to reproduce it): Incorrect version checking in configuration template files. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional N/A

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: N/A Observed behavior: N/A Expected behavior: No errors Proposed solution (Optional): 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 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",'Update version checking template
ENGR00366778	Defect	[WDG] Missing MemMap declarations in driver files 'Problem detailed description (how to reproduce it): The following variables do not have consistent memory section guarding for definition and all other extern declarations: Wdg_au8Index Wdg_pConfigPC the variables shoudl be allocted in the same MemMap.h section where it is defined, and in every other place where it's declared external. this causes a link error when sda=all compiler option is used. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: NA Observed behavior: A link error is reported when sda=all compiler option is used. Expected behavior: Consistent allocation to memory for declaration and definition of variables. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Place both the declaration and the definition of variables inside the same memory section.",'Add MemMap declarations in Dio.c and Dio_Cfg.h
ENGR00373868	NewWork	[WDG] Move clock value of Wdg to structure data. 'NewWork Description: New implement concering Wdg internal clock was introduced since from

Table continues on the next page...

ID	Subtype	Headline and Description
		RaceRunner Ultra release. Wdg clock value must be moved into struct data. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): remove all WDG_FREQUENCY_U16 put (uint16)[!"node:value('./../../WdgGeneral/WdgClockValue')"!] into config struct","'remove all WDG_FREQUENCY_U16 put (uint16)[!"node:value('./../../WdgGeneral/WdgClockValue')"!] into config struct
ENGR00368363	NewWork	[WDG] Move the WDG clock frequency to the mode settings and use the Mcu reference clock 'Initial Description: 1.For some platforms(Kinetis) WDG has various clock sources. The WDG clock must be moved to the WDG mode settings in the xdm and data types to allow to the user to use different clock sources in different modes. 2.For the WDG clock add an interface with MCU: use MCU reference clock. (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...)","'1.For some platforms(Kinetis) WDG has various clock sources. The WDG clock must be moved to the WDG mode settings in the xdm and data types to allow to the user to use different clock sources in different modes. 2.For the WDG clock add an interface with MCU: use MCU reference clock.",ENGR00376775 ENGR00376934

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00372866	NewWork	[WDG] Rename configuration files 'NewWork Description: The post build configuration files must Wdg_PBcfg.c/.h instead of Wdg_PBCfg.c/.h. Otherwise there might be issues with ant_generator.xml. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Rename configuration files Wdg_PBCfg.c/.h to Wdg_PBcfg.c/.h, with lower case "c".','Renamed configuration files.
ENGR00372707	Defect	[WDG] Replace hard-coded copyright to M4_<>_COPYRIGHTED_TO 'Problem detailed description (how to reproduce it): Plugin files shall not contain hard-coded copyright. Proposed solution (Optional): Replace hard-coded copyright with M4_<>_COPYRIGHTED_TO tags.','"<Description of changes, e.g. Check was added to ... The function xyz was modified because ...> Files Modified: <e.g. Adc_Irq.c > <...>
ENGR00378829	Defect	[WDG] The compiler warning when WdgMaxTimeout=65.335 'Problem detailed description (how to reproduce it): In the customer configuration with WdgMaxTimeout = 65335, the following warning is reported by compiler. plugins\Wdg_TS_T40D3M8I1R0\src\Wdg_Channel.c, 937: warning: comparison is always false due to limited range of data type [-Wtype-limits] Preconditions: WdgMaxTimeout = 65335 Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] 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): [...]",'Change is made to make sure no comparison is always true/false to avoid a compiler warning about comparison is not necessary

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00370122	NewWork	[WDG] Update Base address array to display all HW Units 'NewWork Description: The Base Address array in every IP does not display all the base addresses for all the Hardware Units in a platform. In this case, if a platform does not have a specific HW, the offset of all the other HW Units will be shifted. Eg: On DSPI module, we have the following Base Addresses: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #endif #ifdef DSPI2_BASEADDR DSPI2_BASEADDR, #endif ..... If on a platform, the DSPI0 is missing, than the DSPI0 will not be present in the above array. Due to this scenario, the offset of DSPI1 will be 0(zero) instead of 1(one), and it will generate an exception(IVOR) or an incorrect HW Unit will be used. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the base Address array to display all the HW Units. If a Unit will not be present on a platform, the adress 0x0 will be generated: CONST(uint32, SPI_CONST) Dspi_au32BaseAddrs[] = { #ifdef DSPI0_BASEADDR DSPI0_BASEADDR, #else ((uint32)0x00000000UL), #endif #ifdef DSPI1_BASEADDR DSPI1_BASEADDR, #else ((uint32)0x00000000UL), #endif .....",'Updated definition of base addresses.
ENGR00373407	NewWork	[WDG] Use "READONLY" instead of "EDITABLE" in xdm file 'Initial Description: Tresos does not throw a warning if "editable" is set to "false" and the configured value is different to its default value. This could happen if the user change values with an normal text editor and not with tresos Studio. If instead of "editable=false" "readonly=true" would be the better choice for these parameters. EDITABLE:

Table continues on the next page...

ID	Subtype	Headline and Description
		Used to allow the user to edit the element or not. If this attribute value is set to false, the check box will be greyed out. <a:da name="EDITABLE" value="false"/> READONLY: Used to allow the user to edit the element or not. If this attribute value is set to "true", the field will be greyed out and reset to its default value. A warning is shown if the value is changed via other means (e.g. if the internal .xdmfile is edited directly). <a:da name="READONLY" value="true"/> Tresos could verify it and would correct it to the default value and would throw a warning if it was changed. It would be more user friendly.","'Use "READONLY" instead of "EDITABLE" in xdm file
ENGR00378362	NewWork	[WDG] Verify all Misra errors and comments 'NewWork Description: Analyze AUTOSAR_MCAL_<driver>_MisraReport.xlsx document and check MISRA errors and comments associated with the errors. All MISRA errors need to be fixed. There are several rules that can not be fixed and therefore commented out. For details see the attachment MISRADeviations.xls Expected behavior: Fix all MISRA errors that are not in in MISRADeviation.xls Output expected: NA Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]",'Verify all Misra errors and comments
ENGR00374729	Defect	[WDG] Wdg_SetMode and Wdg_SetTriggerCondition cannot be reached via function pointer 'Problem detailed description (how to reproduce it): Issue Type: Bug Issue classification: low(C) Priority: urgent Package: Wdg S32K14X Found in version: MCAL4_0_BETA_0_9_0 What happens (symptoms): ---------- Vector WdgIf calls Wdg API function Wdg_SetMode via function pointer, but it cannot be reached, because the Wdg driver uses function like macro to redefine Wdg_SetMode to Wdg_ChannelSetMode: #define Wdg_SetMode(Mode) Wdg_ChannelSetMode(Mode, WDG_IPW_INSTANCE0) According BSW00330 in AUTOSAR_SWS_WatchdogDriver.pdf (Page47, 12: Not applicable requirements) the requirement for usage of macros / inline function instead of functions has been skipped and would not be allowed.

ID	Subtype	Headline and Description
		When does this happen: ---------- When WdgIf tries to locate Wdg_SetMode() API. In which configuration does this happen: ---------- Any 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): [...]","'Vector WdgIf calls Wdg API function Wdg_SetMode via function pointer, but it cannot be reached, because the Wdg driver uses function like macro to redefine Wdg_SetMode to Wdg_ChannelSetMode: #define Wdg_SetMode(Mode) Wdg_ChannelSetMode(Mode, WDG_IPW_INSTANCE0)

4.5 RTM 1.0.1

ID	Subtype	Headline and Description
ENGR00361523	Defect	[ADC] Conversion results are incorrect for a groups configured with 256 channels. Problem detailed description (how to reproduce it): In the configuration box AdcGroupDefinition: Asssigned 256 channels: From index 0->255 channel 0 is assigned Index 256: Channel 1 is assigned. So from this configuration there will be wrong in somewhere:such as CONST(Adc_ChannelType , ADC_CONST) AssignedChannelCount; Test Case ID (internal TC that caught the defect) - optional Adc_TC_0169
ENGR00361042	Defect	[ADC] 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.

Table continues on the next page...

ID	Subtype	Headline and Description
		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
ENGR00361294	NewWork	[ADC] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a>
ENGR00362085	Defect	[ADC] Fix misra error Fix misra error from Adc.c
ENGR00360025	NewWork	[ADC] Improvement: Move all specific implementation from generate files to resource files NewWork Description: Currently, some code in generate files are specific for each platform. They should be move to resource files, make generate files generic to reduce

Table continues on the next page...

ID	Subtype	Headline and Description
		integration/porting effort.
ENGR00358288	NewWork	[ADC] Review and implement changes according to Errata for Calypso RTM1.0.1 Description: - Review Errata cut 2.0 MPC5748G_1N81M Mask Set Errata Rev .1, 12 May 2015 for Calypso 6M http://compass.freescale.net/go/231283702 - Implement errata workarounds - Fill the review result to the template
ENGR00355965	Defect	[ADC] 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.
ENGR00360269	NewWork	[BASE] Add base address for second ENET NewWork Description: Calypso support 2 ENET. However, in Reg_eSys.h, there is only 1 address for ENET_0. Please add the address for second adress for ENET_1. Please check also other platforms which also have more than one controllers (at the moment, only Calypso has 2 controllers) Expected behavior: Has the following define: #define ENET_0_BASEADDR ((uint32)0xFFCC8000UL) Proposed solution (Optional): add the following define to Reg_eSys.h #define ENET_1_BASEADDR ((uint32)0xFFCC8000UL)
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'

Table continues on the next page...

ID	Subtype	Headline and Description
		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
ENGR00358458	NewWork	[BASE] Define macros for e7780 in Calypso 6M Description: This errata below need to be implemented workaround for Calypso e8770 : FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled (Initial revision) If the FlexRay module is configured in Dual Channel mode, by clearing the Single Channel Device Mode bit (SCM) of the Module Control register (FR_MCR[SCM]=0), and Channel A is disabled, by clearing the Channel A Enable bit (FR_MCR[CHA]=0) and Channel B is enabled, by setting the Channel B enable bit (FR_MCR[CHB]=1), there will be a missing transmit (TX) frame in adjacent minislots (even/odd combinations in Dynamic Segment) on Channel B for certain communication cycles. Which channel handles the Dynamic Segment or Static Segment TX message buffers (MBs) is controlled by the Channel Assignment bits (CHA, CHB) of the Message Buffer Cycle Counter Filter Register (FR_MBCCFRn). The internal Static Segment boundary indicator actually only uses the Channel A slot counter to identify the Static Segment boundary even if the module configures the Static Segment to Channel B (FR_MBCCFRn[CHA]=0 and FR_MBCCFRn[CHB]=1). This results in the Buffer Control Unit waiting for a corresponding data acknowledge signal for minislot:N in the Dynamic Segment and misses the required TX frame transmission within the immediate next minislot:N+1. Proposed solution Add errata define: ERR_IPV_FLEXRAY_0005 to Base module
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.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00355454	Defect	[CAN] Collect comments are reported on Panther 0.9.1 and Panther 1.0.0 Initial Description: Customer is sent these comments. (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): Please take a look file detail in attachment Preconditions: Panther 0.9.1 Panther RTM 1.0.0 Reported release baseline: BLN_MCAL_4.0_PANTHER_BETA_0.9.1 BLN_SMCAL_4.0_PANTHER_RTM_1.0.0 Proposed solution (Optional): Review driver source Expected behavior: Verify these comments, if they are existing in driver source code please plan to fix them They are ready fixed in our source please fill CR or baseline into attachment file Requirement source: Customer Request
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)"!] [!IF "num:i(CanControllerId) = num:i($x)"!] [!VAR "maxmb" = "$maxmb+1"!] [!ENDIF!] [!ENDSELECT!] [!ENDLOOP!]
ENGR00356454	Defect	[CAN] Fix error relate attribute EDITABLE

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): Users can not configure CanMainFunctionReadPeriod/ CanMainFunctionWritePeriod due to the value of attribute EDITABLE is "false" Test Case ID (internal TC that caught the defect) - optional TC0063 Observed behavior: Configurations of CanMainFunctionReadPeriod/CanMainFunctionWritePeriod Expected behavior: Users can configure CanMainFunctionReadPeriod/ CanMainFunctionWritePeriod parameters Proposed solution (Optional): Set attribute EDITABLE to "true"
ENGR00357735	Defect	[CAN] Incorrect clear Bus Off flag when other interrupt occur Problem detailed description (how to reproduce it): In platform have 1 ISR for many controller. When have an interrupt, Bus Off flag will be clear. Preconditions: CAN driver is in busoff state (BOFFINT bit is set), Can BusOff Processing Type is polling. Test Case ID (internal TC that caught the defect) - optional Can_TC_1001 Proposed solution (Optional): update the code as below: if (NULL_PTR != Can_pCurrentConfig) \ { \ if (0U != can_temp) \ { \ /* @violates @ref Can_Flexcan_h_REF_3 Violates MISRA 2004 Advisory Rule 17.4, pointer arithmetic other than array indexing used */ \ if (CAN_OK == Can_IPW_SetControllerMode(CAN_FC##FC##_INDEX, &(Can_pCurrentConfig->ControlerDescriptors[CAN_FC##FC##_INDEX]), CAN_T_STOP,(boolean)FALSE)) \ { \ /* Process BusOff condition for controller ID of FlexCAN FC */ \ Can_IPW_ProcessBusOff(CAN_FC##FC##_INDEX); \ CanIf_ControllerBusOff(CAN_FC##FC##_INDEX); \ } \ /* Clear the ESR[BOFF_INT] bus off interrupt flag (w1c). */ \ /* @violates @ref Can_Flexcan_h_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ \ /* @violates @ref Can_Flexcan_h_REF_3 Violates MISRA 2004 Advisory Rule 17.4, pointer arithmetic other than array indexing used */ \ REG_WRITE32(FLEXCAN_ESR(FLEXCAN_##FC##_OFFSET), FLEXCAN_ESR_BOFFINT_U32) ; \ } \ } \
ENGR00357350	Defect	[CAN] Incorrect section for configuration pointer Problem detailed description (how to reproduce it): The following configuration pointers (Can_pCurrentConfig and Can_pControlerDescriptors) are initialized in declaration. However, the current

Table continues on the next page...

ID	Subtype	Headline and Description
		implementation put them in CAN_START_SEC_VAR_NO_INIT_UNSPECIFIED which does not properly. Expected behavior: These variables should be in section: CAN_START_SEC_VAR_INIT_UNSPECIFIED
ENGR00358571	Defect	[CAN] RxFifoGlobalMask generate is not correct when CAN_RXFIFO_ENABLE = ON Problem detailed description (how to reproduce it): Customer question: Continental is reporting an issue with the indexing of CAN hardware objects. Attached is their configuration file (Can.xdm) -> in file Can_Flexcan.c ->in function Can_FlexCan_ProcessRx(...) -> the u32MbHrh is not computed correctly for the received CanMsgIds before passing it to CanIf layer, when calling function CanIf_RxIndication(u32MbHrh, u32MbMessageId, u8MbDataLength, u8CanMbData); We also saw that the compiler optimization influence this behavior, but it has a random behavior. (flag -Xopt-count) I would expect that the u32MbHrh to contain the index in the following table for each message : /* Array containing RX PDUs Configuration */ CONST(CanIf_tstRxPduConfig, CANIF_CONST) CanIf_RxPduList [9] = { { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpEtcSafeProtFr02, /* UserPduId */ 0x120u, /* CanId */ 8u, /* Dlc */ 0u, /* RxFuncIndex */ }, /* [0] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr02, /* UserPduId */ 0x57Fu, /* CanId */ 4u, /* Dlc */ 0u, /* RxFuncIndex */ }, /* [1] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr06, /* UserPduId */ 0x100u, /* CanId */ 8u, /* Dlc */ 0u, /* RxFuncIndex */ }, /* [2] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr10, /* UserPduId */ 0x80u, /* CanId */ 8u, /* Dlc */ 0u, /* RxFuncIndex */ }, /* [3] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr12, /* UserPduId */ 0xD0u, /* CanId */

Table continues on the next page...

ID	Subtype	Headline and Description
		8u, /* Dlc */
		0u, /* RxFuncIndex */ }, /* [4] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr13, /* UserPduId */ 0xE0u, /* CanId */ 8u, /* Dlc */ 0u, /* RxFuncIndex */ }, /* [5] */ { PduRConf_PduRSrcPdu_PduRSrcPdu_Rx_ptpAsdmSafeProtFr14, /* UserPduId
ENGR00355970	Defect	[CAN] 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.
ENGR00355530	Defect	[DIO] Correct XPath expression problem in template file and support reversed bits in Read/Write Channel Group functions Initial Description: Customer comment: The XPath expression in the following conditional statement is incorrect: [!IF "../../../../DioGeneral/DioReversePortBits"!]. In the context where this it is evaluated the correct expression is ../../../../../../DioGeneral/DioReversePortBits (Note: After the CR type is decided, will remain only one section from the two listed below, the other one will be deleted) Reported release baseline: BLN_MCAL_4.0_PANTHER_BETA_0.9.1
ENGR00361295	NewWork	[DIO] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected

Table continues on the next page...

ID	Subtype	Headline and Description
		are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a> Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00361305	NewWork	[DIO] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00355966	Defect	[DIO] The short name of Std_Types is Std but not STD_TYPES Problem detailed description (how to reproduce it): According to AUTOSAR_TR_BSWModuleList document, the short name(API service prefix) of Std_Types.h is 'Std', which leads to

Table continues on the next page...

ID	Subtype	Headline and Description
		STD_AR_RELEASE_MAJOR_VERSION being defined instead STD_TYPES_AR_RELEASE_MAJOR_VERSION. Need to replace STD_TYPES_AR_RELEASE_MAJOR_VERSION with STD_AR_RELEASE_MAJOR_VERSION in all places where a check on the version of Std_Types.h file is happening.
ENGR00361178	Defect	[ETH] 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
ENGR00361600	Defect	[ETH] Correct misra violations Problem detailed description (how to reproduce it): There are misra violations in the latest label. Please correct or comment them. Expected behavior: Able to generate MISRA reports without error Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Fix and comment all violations reported
ENGR00361297	NewWork	[ETH] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in

Table continues on the next page...

ID	Subtype	Headline and Description
		Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a>
ENGR00361800	Defect	[ETH] Eth_SetPhysAddr declaration violates the Autosar declaration According to Autosar SWS for the Eth Driver the function declaration for the API Eth_SetPhysAddr Req:SWS_Eth_00151 is void Eth_SetPhysAddr(uint8 CtrlIdx, const uint8* PhysAddrPtr) while the actual code is void Eth_SetPhysAddr(uint8 CtrlIdx, uint8* PhysAddrPtr); This casues an issue since the function EthIf_SetPhysAddr calls the driver function Eth_SetPhysAddr with a paramter P2CONST not P2VAR. Please fix the in function Eth_SetPhysAddr and all other internal function that is called from this function Eth_Enet_SetPhysAddr Eth_Ipw_SetPhysAddr
ENGR00355408	Defect	[ETH] Fix misra error Problem detailed description (how to reproduce it): CR ENGR00355142 implement another solution for prevent misra error cause by Det API. However, the solution is different with other module. Please implement the solution that other module applied.
ENGR00356879	Defect	[ETH] Possible issue when configure DEM and loopback with new xdm template Problem detailed description (how to reproduce it): When remove ENABLE attribute, there is problem with the OPTIONAL node which also has ENABLE=false. This cause issue unable to modified the value of this node. Expected behavior: Able to change value of optional node.
ENGR00357960	NewWork	[ETH] The Eth driver supports multiple Eth controllers Currently the Eth driver only supports maximum one controller. In some platforms (MPC5748G) have more than one controller (e.g gateway application), the customer needs to use more than one controller in an ECU but MCAL Eth driver does support only one controller.

Table continues on the next page...

ID	Subtype	Headline and Description
		Is it possible to support multiple controllers in Eth driver?
ENGR00361307	NewWork	[FEE] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00361167	NewWork	[FEE] Replace STATIC with static(lowercase) NewWork Description: STATIC definition was removed from Base, FEE should use the C keyword static instead.
ENGR00355975	Defect	[FEE] 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.
ENGR00360043	Defect	[FLS] Add #ifdef guard for cache invalidation macros Problem detailed description (how to reproduce it): Warning is issued for undefined macro CACHE_INVALIDATE_MACROS. Proposed solution (Optional): Add #ifdef guard for every cache invalidate macros, in order to avoid compiler warnings for platforms where CACHE_INVALIDATE_MACROS is not defined.
ENGR00360976	Defect	[FLS] Avoid multiple definition of STATIC macro

Table continues on the next page...

ID	Subtype	Headline and Description
		Mcal.h ========== 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
ENGR00356154	Defect	[FLS] The short name of MemIf_Types is MemIf but not MEMIF_TYPES Problem detailed description (how to reproduce it): The short name of MemIf_Types is MemIf but not MEMIF_TYPES Proposed solution: MemIf_Types.h file in MEMIF module will change its macros for version checking from MEMIF_TYPES_AR_RELEASE_MAJOR_VERSION to MEMIF_AR_RELEASE_MAJOR_VERSION MEMIF_TYPES_AR_RELEASE_MINOR_VERSION to MEMIF_AR_RELEASE_MINOR_VERSION MEMIF_TYPES_AR_RELEASE_REVISION_VERSION to MEMIF_AR_RELEASE_REVISION_VERSION . All other modules that include this file must adjust their version checking code for this accordingly.
ENGR00355976	Defect	[FLS] 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.
ENGR00361082	Defect	[FR] 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
ENGR00361300	NewWork	[FR] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a> Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00359494	NewWork	[FR] Create quality package for Calypso 3M/6M RTMv1.0.1 NewWork Description: Create quality package for Calypso 3M/6M RTM v1.0.1 with below:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Test summary report - MISRA, VSMD - LDRA report - Compiler warning - Profiling, code stack size report - Traceability Matrix Note: - The only reports that we need to duplicate are the test reports, same way we did for Beta/RTM v100 release. For all the others, one report is enough. - Test Specification report and tool will need to be updated to include some information about differences between 3M and 6M testing. - Will run profiling and code stack size on 3M chip (cut 1.0, 324 pins). If for some reason, profiling on 3M does not work, we will change the plan and generate Profiling + Code Stack on Calypso 6M. - All the other reports will be generated form 6M station, with 6M name.
ENGR00355517	Defect	[FR] Problem while parsing code template file Initial Description: Customer comment: The comment in the following line prevents our parser from correctly parsing this template: [!ELSE!][!// "text:tolower(node:value(node:ref($FrController)/VendorSpecific/SingleChannelModeEnabled))='false'"!][!// Please change this line to [!ELSE!][!// to fix the problem without affecting the generated 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): N/A Preconditions: BLN_SMCAL_4.0_PANTHER_RTM_1.0.0 Observed behavior: N/A When can it be observed? (at configuration time, at runtime, at compile time?) N/A Expected behavior: N/A Reported release baseline: N/A Proposed solution (Optional): N/A NewWork Classification: (internal task, improvement, feature request) N/A In case for feature request, does a TWG ticket exist (cloned or linked)? TBC (Yes/No) NewWork Description: N/A Expected behavior: N/A Requirement source: Customer request

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00358294	NewWork	[FR] Review and implement changes according to Errata for Calypso RTM1.0.1 Description: - Review Errata cut 2.0 MPC5748G_1N81M Mask Set Errata Rev .1, 12 May 2015 for Calypso 6M http://compass.freescale.net/go/231283702 - Implement errata workarounds - Fill the review result to the template
ENGR00355977	Defect	[FR] 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.
ENGR00355515	Defect	[FR] Wrong argument type of node:path() function in template file Initial Description: Customer comment: In Fr's templates: In the following conditional statement node:value is used instead of node:ref to get the node referred by the specified reference node, then passed to node:path, which expects a Node, while node:value returns a String: [!IF "$FrCtrlPath = node:path(node:value(FrIfFrCtrlRef))"!] The correct statement would be: [!IF "$FrCtrlPath = node:path(node:ref(FrIfFrCtrlRef))"!] (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): N/A Preconditions: N/A Observed behavior: N/A When can it be observed? (at configuration time, at runtime, at compile time?) N/A Expected behavior: N/A Reported release baseline: BLN_MCAL_4.0_PANTHER_BETA_0.9.1

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): N/A NewWork Classification: (internal task, improvement, feature request) N/A In case for feature request, does a TWG ticket exist (cloned or linked)? TBC (Yes/No) NewWork Description: N/A Expected behavior: N/A Requirement source: Customer Request
ENGR00361298	NewWork	[GPT] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a> 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
ENGR00361310	NewWork	[GPT] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00360842	Defect	[GPT] GptChannelTickValueMax validator is incorrect Problem detailed description (how to reproduce it): The max value for EMIOS channel seems to be incorrect. It would be 65536 since the 1 tick is subtracted from the inputted value in the template code. This is also keep the consistency with STM/PIT/RTC where the max value is 4294967296
ENGR00357920	Defect	[GPT] MCAL / some files exist doubled eMios_Common_Types.h (Gpt, Pwm, Mcl) Siul2_IPVersion.h (Port, Dio) Reg_eSys_eMios.h (Pwm, Mcl) Customer complains that these files are delivered twice within MCAL delivery package (see module short names in brackets). Please remove doubled files which are not intended. This causes confusion on Customer side (in best case) and problems (in the worst case)
ENGR00357994	NewWork	[GPT] Set dual clock mode features NewWork Description: RTC Timer can do exactly in dual clock mode when use set clock mode function: Setup RTC_Prescaler or RTC_Prescaler_Alternate in EB tresos, this parameters will be used to dived clock source which the tests case set clock mode in Normal or sleep Expected behavior: Dual clock mode features works correctly. Proposed solution (Optional): Update: - Gpt_Ipw.c - Gpt_PluginMacros.m - Gpt_Ipw_Types.h to add dual clock mode for RTC

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00355504	Defect	[GPT] Syntax error and version check is missing in template files Problem detailed description (how to reproduce it): 1. At some places the ERROR statement contains an unnecessary space, which makes it syntactically incorrect: [!ERROR !]. 2. In module configuration templates do not check if the version number in the configuration (stored in CommonPublishedInformation) matches the one hard-coded into the templates 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: 1. Update syntax error 2. Add to check version in template files Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): To check version in template files, my suggest as you can use as other modules for example Flexray driver. They has checked version in Fr_VersionCheck.m after that include this check into template files
ENGR00355978	Defect	[GPT] 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.
ENGR00361047	Defect	[GPT] Update definition of STATIC marco Problem detailed description (how to reproduce it): STATIC macro in base module is redefined by static macro. Driver have to be updated to use standard C keyword "static" instead Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: [...] Observed behavior: [...] Expected behavior: Use static macro instead STATIC macro 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 macro by static macro in all files
ENGR00358397	Defect	[GPT] Update xdm file for GptChannelTickValueMax parameter Problem detailed description (how to reproduce it): the bit width of eMios timer is 16 bits so the GptChannelTickValueMax value is 65536. Gpt_ValidateParamValue function is failed if GptChannelTickValueMax value is wrong. Expected behavior: GptChannelTickValueMax value is suite for 16 bits for eMios timer. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update GptChannelTickValueMax value in Gpt.xdm file
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
ENGR00360295	NewWork	[ICU] Change define for making test NewWork Description: Change define for using EMIOS: from #define EMIOS to #define EMIOS_USED. Expected behavior: The old define affected to some platform.
ENGR00351901	Defect	[ICU] Counter bus is not set up for ICU channel

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]
ENGR00361311	NewWork	[ICU] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00360751	NewWork	[ICU] Fix Misra violations for Calypso RTM 1.0.1 NewWork Description: Fix Misra violations for Calypso RTM 1.0.1 Expected behavior: No Misra violations
ENGR00355979	Defect	[ICU] 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 .

Table continues on the next page...

ID	Subtype	Headline and Description
		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.
ENGR00354954	NewWork	[ICU] Wakeup capable is enabled and ICU channel in "ICU_MODE_TIMESTAMP" mode Hello, Bellow as customer's request about able to configure ICU channel in "ICU_MODE_TIMESTAMP" and enable Wakeup capable: "With the Bolero MCAL we are able to configure the ICU channel in "ICU_MODE_TIMESTAMP" and wakeup capability enabled. But with the Calypso MCAL, we are not able to configure the ICU channel in "ICU_MODE_TIMESTAMP" with wakeup capability enabled." CE comments: We can able to enable Wakeup capable and set ICU channel as "ICU_MODE_SIGNAL_EDGE_DETECT" and other measurement mode (ICU_MODE_EDGE_COUNTER, ICU_MODE_SIGNAL_MEASUREMENT, ICU_MODE_TIMESTAMP) after set ../IcuNonAUTOSAR/ IcuAllowWkpOnAllChls = 'true' in Bolero MCAL So please analysis this functional on Calypso MPC574xG MCAL 0.9.0
ENGR00355184	Defect	[ADC] DMA transfer is configured for incorrect Group in some use cases Problem detailed description (how to reproduce it): DMA transfer is configured for incorrect Group in some use cases (SW one shot group with limit checking failed still pending in the queue). Adc_Adcdig_StartDmaOperation needs to have added a Group parameter, otherwise it will always take the first Group in SW queue for DMA configuration and the HW Group will not have DMA configured properly.
ENGR00359716	Defect	[ADC] Group with BCTU trigger is ignored Problem detailed description (how to reproduce it): The ISR (JECH) is not raised when group is configured as BCTU trigger. This cause failing in checking condition to invoke function Adc_Adcdig_DmaEndHardwareConv.
ENGR00361881	Defect	[ADC] After finish conversion all channels in a list, the destination address does not setting correctly. Problem detailed description (how to reproduce it): In the function Bctu0_AdcDmaEndConversion after all data are transferred to the buffer Adc_u32BctuDmaInternalBuffer, the destination address for all DMA

Table continues on the next page...

ID	Subtype	Headline and Description
		channels did not re-set correctly. Here there is only the destination of an DMA channel is reset. How to produce: -BCTU is configured with Multiple Parallel Conversions mode - ADC0 and ADC1 is configured. - Trigger BCTU such that the list is converted at least two times (BCTU is trigger more than two times). Test Case ID (internal TC that caught the defect) - optional Adc_TC_0190
ENGR00361850	Defect	[ADC] Conversion results are incorrect for a groups configured with 256 channels. Problem detailed description (how to reproduce it): In the configuration box AdcGroupDefinition: Asssigned 256 channels: From index 0->255 channel 0 is assigned Index 256: Channel 1 is assigned. So from this configuration there will be wrong in somewhere:such as CONST(Adc_ChannelType , ADC_CONST) AssignedChannelCount;
ENGR00355983	Defect	[MCU] 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.
ENGR00359847	NewWork	[MCL] Improve definition of the DMA base addresses NewWork Description: Improve definition of the DMA base addresses. Proposed solution (Optional): In Reg_e_Sys_DMa.h improve definition of the DMA base adresses.
ENGR00360795	NewWork	[MCL] Remove compiler warning regarding not used parameter NewWork Description: Remove compiler warning regarding not used parameter.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00361079	Defect	[SPI] Avoid multiple definition of STATIC macro Mcal.h ========== 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
ENGR00360528	Defect	[SPI] Fix Misra error Problem detailed description (how to reproduce it): We have the Misra error in line 1777, rule 10.1. Please see the attachment Expected behavior: Misra error not appear Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
ENGR00362202	Defect	[SPI] Fix misra error Problem detailed description (how to reproduce it): Fix the Misra rule 11.1 in file Spi_DSPI.c line 1812 Expected behavior: No misra appear
ENGR00358100	Defect	[SPI] Keep the Chip select in whole Job transmission The customer report that the Chip select is release at the end of each channel in a Job. However they expect that the chip select is asserted in whole job basis. We think that AUTOSAR requires Chip Select handling according our expectations. Therefore please consider AUTOSAR_SWS_SPIHandlerDriver V3.2.0 R4.0 Rev3: - Chapter 2 (Acronyms and abbreviations) defines meaning of 'Job': A Job is composed of one or several Channels with the same Chip Select (is not released during the processing of Job). A Job is considered atomic and therefore cannot be interrupted by another Job. A Job has an assigned priority. - [SPI066] The Chip Select (CS) is attached to the Job definition. - [SPI263] Chip Select shall be handled during Job transmission and shall be released at the end of it. This Chip Select handling shall be done according to the Job configuration parameters. The customer configuration is in attachment tab. CE's comment: There is a similar limitation-ENGR00347181 but as mentioned above we need to improve this behavior. As my observation, the is caused by using EOQ in the SPI command to raise interrupt for update FIFO or CTAR. The issue could be resolved by using RFDF instead of EOQ. However for the

Table continues on the next page...

ID	Subtype	Headline and Description
		long channel with fast baud-rate the interrupt rate may be very high.
ENGR00361629	Defect	[SPI] The incorrect data to send when driver in Slave mode Problem detailed description (how to reproduce it): The Master will receive the first data with duplicated and frame will be missed last byte. Test Case ID (internal TC that caught the defect) - optional Spi_TS_023 Spi_TS_028 Expected behavior: The Master will receive correct data Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
ENGR00356093	Defect	[MCL][ICU][PWM][GPT] 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.
ENGR00354955	Defect	[PWM] Spikes pulses occur when driver is initialized and when idle mode is exited Following configuration for the related output: PwmChannelClass: PWM_VARIABLE_PERIOD PwmHwChannel: EMIOS_0_CH_4 PwmPolarity: PWM_LOW PwmDutyCycleDefault: 0x8000 (100%) PwmModeSelect: PWM_MODE_OPWMCB_LEAD_DEADTIME EmiosUnifiedChannelBusSelect: PWM_BUS_DIVERSE PwmOffset: 0 PwmTriggerDelay: 0 Pwm_DeadTime: 0 OffsetDelayAdjust: FALSE FreezeEnable: TRUE *** Preconditions: uC in Reset State + run through Pwm_Init() *** Observed behavior: glitch (around 22us) on the output pin

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution Spike pulse also generate in some other modes (OPWMT, DAOC, OPWFMB). To avoid glitch, in SetReg_mode function, check that if come from GPIO mode, flowing steps must be executed: - If dutycycle = 100%, when enter mode, EDPOL bit get complement of polarity after that, EDPOL bit is restored valid value - Else, polarity and mode bits is writen into Cn simultaneously
ENGR00360733	Defect	[PWM] fix misra and compiler warning misra: File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\include\Pwm_eMios_Types.h", line 260, MISRA Rule Violated 5.1 (Required): no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\src\Pwm_eMios.c", line 1836, MISRA Rule Violated 10.1 (Required): no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\src\Pwm_eMios.c", line 1836, MISRA Rule Violated 10.1 (Required): no MISRA violation comment was found (maybe wrong format is used). File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\src\Pwm_eMios.c", line 2206, MISRA Rule Violated 1.2 (Required): no relevant comment was found for MISRA violation on this line. File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\src\Pwm_eMios.c", line 2211, MISRA Rule Violated 1.2 (Required): no relevant comment was found for MISRA violation on this line. File "c:\vv_tools\eb\EB_tresos_Studio_14.2.1_b140128-1223\plugins \Pwm_TS_T2D35M10I1R0\src\Pwm_eMios.c", line 2221, MISRA Rule Violated 1.2 (Required): no relevant comment was found for MISRA violation on this line.
ENGR00361093	Defect	[PWM][ICU][GPT] 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.
ENGR00339846	NewWork	[ETH] -Implements workaround for hardware cache issue NewWork Description: Implements software workaround in order to solve the hardware issue which prevents to define specific memory regions as not cache able. (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Flush cache after any write access to the memory. Invalidate cache before any read access the memory.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00358049	Defect	[FLS] Incorrect Fls_Flash_NumberOfStepBack calculation in Async mode The "Fls_Flash_SectorWrite" function uses the variable "Fls_Flash_NumberOfStepBack" to keep track of the number of bytes written in one operation, in order to read back the values if FLS_WRITE_VERIFY_CHECK == STD_ON. Problem description: On the "Async" branch of the "Fls_Flash_SectorWrite" function, for the unaligned access case(page writes), the variable is calculated as: Fls_Flash_NumberOfStepBack = Fls_ConfigPtr->sectorProgSize[Fls_JobSectorIt]/4U; instead of: Fls_Flash_NumberOfStepBack = Fls_ConfigPtr->sectorPageSize[Fls_JobSectorIt]/4U; Because of this, when unaligned writes(writes made with less bytes than the configured "Programming size") are made in "Async" mode and "Fls Write Verify Check" is ON, the check might fail or a different than configured flash area may be accessed. Proposed solution: Change value to "Fls_ConfigPtr->sectorPageSize[Fls_JobSectorIt]/4U".
ENGR00350501	NewWork	[CAN] Add support for Fault Injection Test point NewWork Description: Add support in MCAL drivers for Fault Injection Test points. Details in attached presentation and on page: http://zcz01web02:1080/wiki/vav/Wiki%20Pages/TestDesignWithFaultInjection.aspx Expected behavior: Updated drivers to add fault injection points Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): -Add in Base the following defines: PUT_IN_QUOTES, MCAL_FAULT_INJECTION_POINT -> they should be empty in case fault injection tests are not compiled (using define ?d ENABLE_FAULT_INJECTION) -In test files (tickets for ITG) In the fault injection tests make files we need to have ?d ENABLE_FAULT_INJECTION -In driver code (tickets for developers) Add label in driver code (tester and developer should agree on the label name and the place in the code) MCAL_FAULT_INJECTION_POINT (LABEL_NAME); Note: LABEL_NAME should be compliant to the following naming rule: <MDL>_FIP_<COUNT>_<COMMENT> Eg: FLS_FIP_1_UPDATE_VARIABLE_A
ENGR00361428	Defect	[CAN] Avoid multiple definition of STATIC macro

Table continues on the next page...

ID	Subtype	Headline and Description
		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
ENGR00361880	Defect	[CAN] Can't receive MB have type "FULL" when BCC enable. Problem detailed description: The reception is not success when BCC enable and type of MB is FULL. The first analysis: The cause can as-synchronize between ID filter mask and number of filter mask configured
ENGR00358260	NewWork	[CAN] Comment or resolve warnings in data flow anomalies Initial Description: Check and resolve dataflow anomalies as much as possible. (Note: After the CR type is decided, will remain only one section from the two listed below, the other one will be deleted)
ENGR00360745	Defect	[CAN] DataLength and CanId not consistent in case of overrun with DET on Initial Description: Client use the CAN driver of our MCAL 1.0.0 (Autosar 4.0) with Panther. They have configured their CAN driver to read the mailboxes in polling mode. They observed that on overrun situation, the message that caused the overrun situation is stored in the last mailbox in the list, independently of the ID and the filter configuration. The function Can_FlexCan_ProcessRx() seems to be the cause for this behaviour. See the code extract below, with the >>> <<< showing how the MB index is set to the end of the list in case of overrun. "Can_FlexCan_ProcessRx(....) { ... if(FLEXCAN_MBCS_CODERXOVRR_U32 == (u32MbConfig & FLEXCAN_MBCS_CODE_U32)) { >>> u8MbIndex = mbindex_end; <<< Det_ReportError((uint16)CAN_MODULE_ID, (uint8)CAN_INSTANCE, (uint8)CAN_SID_MAIN_FUNCTION_READ, (uint8)CAN_E_DATALOST); REG_READ32(FLEXCAN_TIMER(u8HwOffset)); } ... u32MbMessageId = ((uint32)0x0U == (u32MbConfig &

Table continues on the next page...

ID	Subtype	Headline and Description
		(uint32)FLEXCAN_MBCS_IDE_U32)) ? /* condition to test */ (Can_IdType)((uint32)((REG_READ32(FLEXCAN_MB_ID((u8HwOffset), (uint32)(>>>u8MbIndex<<<))) & (uint32) (FLEXCAN_MBID_ID_STANDARD_U32)) >> (uint32)(18U))) : /* value if condition is true */ (Can_IdType)((uint32)(REG_READ32(FLEXCAN_MB_ID((u8HwOffset), (uint32)(>>>u8MbIndex<<<))) & (uint32) (FLEXCAN_MBID_ID_EXTENDED_U32)) | FLEXCAN_MBC_ID_IDE_U32); /* value if condition is false */ .... } Is this beviour expected (overun frames to be received in the last MB ?) Are there some recommendations regarding the integration ? For information you can find the configuration file in attachment;
ENGR00355412	Defect	[CAN] Fix Misra error Problem detailed description (how to reproduce it): There is new misra error due to change of DET. Developer will fixed this issue by add comment.
ENGR00357127	Defect	[CAN] RX interrupts occur prior TX interrupts Violate the requirement CPR-MCAL-742.can In case multiple events (e.g., RX and TX interrupts) are simultaneously pending and the actual order of occurrence can no longer be obtained, those events caused by TX interrupts shall be processed and indicated to upper layers prior to the event caused by RX interrupts. Root cause: In the function Can_Isr_A, Can_Isr_B and Can_Isr_C: the events caused by RX interrupts is processed and indicated to upper layers prior to the events caused by TX interrupts. Line 969 ? 973 (in Can_FlexCan.h) Proposed solution: Replace: /* Rx: process from 0 to (FirstTx-1) */ \ Can_IPW_ProcessRx(CAN_FC##FC##_INDEX, (uint8)0U, (uint8) (Can_ControllerStatuses[CAN_FC##FC##_INDEX].u8FirstTxMBIndex - (uint8)1U)); \ /* Tx: process from FirstTx to (MaxMB-1) */ \ /* @violates @ref Can_Flexcan_h_REF_3 Violates MISRA 2004 Advisory Rule 17.4, pointer arithmetic other than array indexing used */ \ Can_IPW_ProcessTx(CAN_FC##FC##_INDEX, (uint8)Can_ControllerStatuses[CAN_FC##FC##_INDEX].u8FirstTxMBIndex, (uint8)(Can_pCurrentConfig- >ControlerDescriptors[CAN_FC##FC##_INDEX].u8MaxMBCount - (uint8)1U)); \ By: /* Tx: process from FirstTx to (MaxMB-1) */ \ /* @violates @ref Can_Flexcan_h_REF_3 Violates MISRA 2004 Advisory Rule 17.4, pointer arithmetic other than array indexing used */ \ Can_IPW_ProcessTx(CAN_FC##FC##_INDEX, (uint8)Can_ControllerStatuses[CAN_FC##FC##_INDEX].u8FirstTxMBIndex, (uint8)(Can_pCurrentConfig- >ControlerDescriptors[CAN_FC##FC##_INDEX].u8MaxMBCount - (uint8)1U));

Table continues on the next page...

ID	Subtype	Headline and Description
		\ /* Rx: process from 0 to (FirstTx-1) */ \ Can_IPW_ProcessRx(CAN_FC##FC##_INDEX, (uint8)0U, (uint8) (Can_ControllerStatuses[CAN_FC##FC##_INDEX].u8FirstTxMBIndex - (uint8)1U)); \
ENGR00359270	NewWork	[CAN] Switch the ISR implementation based on a parameter from resource files NewWork Description: Add support in order to use multiple interrupt lines even the architecture is ARM. Expected behavior: KFA porting from Calypso
ENGR00360760	Defect	[CAN] TX MB interrupt flag cleared by RX processing function if no RX HW object configured Problem detailed description (how to reproduce it): We have the following configuration: CAN controller[0] is used only for transmit purpose. Uses FLEXCAN_1 hardware. CAN controller[1] is used only for receive purpose. Uses FLEXCAN_4 hardware. CAN controller[2] is used for transmit and receive purpose. Uses FLEXCAN_0 hardware. We use polling mechanism and not interrupt. So, we have configured CanTxProcessing = POLLING CanRxProcessing = POLLING To read, we call Can_MainFunction_Read() which in turn calls FlexCan_LLD_Process_Rx(). For CAN controller[0] since there are no receive messages, the function FlexCan_LLD_Process_Rx() has the parameter values as controller = 0, mbindex_start = 0, mbindex_end = 0 There is a condition in the code inside function FlexCan_LLD_Process_Rx() while(can_mb_index <= mbindex_end) and this condition will be satisfied always even if there are no receive buffers configured. This is leading to the situation that it clears the transmit confirmation flag in IFLAG register. In our project, we need confirmation from IFLAG_1[BUF_0]. But we see sometimes that this flag is cleared erroneously by FlexCan_LLD_Process_Rx() because of entering the above while loop. As a temporary workaround, 1. I can try making CanRxProcessing = INTERRUPT for controller[0]. This will be like fooling the CAN module so that it does not call FlexCan_LLD_Process_Rx() 2. Configure at least one RX HW object on each CAN controller Preconditions: No RX HW Object configured for CAN controller. TX and RX processing configured to polling.
ENGR00356883	NewWork	[CAN] Update define FLEXCAN_MCR_MDIS_U32_U32 NewWork Description: Old define ID is FLEXCAN_MCR_MDIS_U32_U32 violate coding rule

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): Change from FLEXCAN_MCR_MDIS_U32_U32 to FLEXCAN_MCR_MDIS_U32
ENGR00357420	Defect	[CAN] Correct the implementation of errata e8341 Problem detailed description (how to reproduce it): Current implementation of errata e8341 done for Rainier platform is not correct because it does not maintain old behavior when it is disabled. This is caused by the fact that the old behavior shall be executed when define ERR_IPV_FLEXCAN_0014 is not defined. Also the implementation shall raise an error when ERR_IPV_FLEXCAN_0014 is defined as STD_OFF Preconditions: Errata e8341 is enabled which means that define ERR_IPV_FLEXCAN_0014 is defined into BASE module Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Can_Write does not return CAN_NOT_OK when the timeout counter expired during a controller mode change. Expected behavior: Can_Write shall return CAN_NOT_OK when the timeout expire during the mode change from NORMAL to FREEZE. 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 code guarded by ERR_IPV_FLEXCAN_0014 into Can_Flexcan.c file as below: #ifdef ERR_IPV_FLEXCAN_0014 #if (ERR_IPV_FLEXCAN_0014 == STD_ON) /* Backup MCR register */ /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ u32TempMCR = (uint32)REG_READ32(FLEXCAN_MCR(u8HwOffset)); /* Backup IMASKn registers */ /* @violates @ref Can_Flexcan_c_REF_8 Violates MISRA 2004 Required Rule 17.4, pointer arithmetic other than array indexing used */ for (u8RegCount1=(uint8)0; u8RegCount1< (uint8) ((uint8)u8MessageBufferControllerSize[Can_pControlerDescriptors[Controller].u8ControllerOffset] >> FLEXCAN_MB_SHIFT5BIT_U8) ;u8RegCount1++) { /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ u32TempIMASK[u8RegCount1] = REG_READ32(Can_IflagImask[u8RegCount1][u8HwOffset].u32CanImask); } /* Set the Soft Reset bit (SOFTRST) in MCR */ /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ REG_BIT_SET32(FLEXCAN_MCR(u8HwOffset), FLEXCAN_MCR_SOFTRST_U32); /* Poll the MCR register until the Soft Reset bit (SOFTRST) bit is cleared */

Table continues on the next page...

ID	Subtype	Headline and Description
		u32TimeoutCount = (uint32)CAN_TIMEOUT_DURATION; /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ while ((FLEXCAN_MCR_SOFTRST_U32 == ((uint32) (REG_READ32(FLEXCAN_MCR(u8HwOffset))) & FLEXCAN_MCR_SOFTRST_U32)) && (u32TimeoutCount > (uint32)0x0U)) { /* MCR[SOFT_RST] still set -> reset in progress. */ u32TimeoutCount--; } /* The soft reset has completed Reconfigure the Module Control Register (MCR) */ /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ REG_WRITE32(FLEXCAN_MCR(u8HwOffset), u32TempMCR); /* Reconfigure all the Interrupt Mask Registers (IMASKn) */ /* @violates @ref Can_Flexcan_c_RE
ENGR00358448	NewWork	[FR]Implement workaround for errata e8770 in Calypso 6M Description: ERR008770 : FlexRAY: Missing TX frames on Channel B when in dual channel mode and Channel A is disabled (Initial revision) If the FlexRay module is configured in Dual Channel mode, by clearing the Single Channel Device Mode bit (SCM) of the Module Control register (FR_MCR[SCM]=0), and Channel A is disabled, by clearing the Channel A Enable bit (FR_MCR[CHA]=0) and Channel B is enabled, by setting the Channel B enable bit (FR_MCR[CHB]=1), there will be a missing transmit (TX) frame in adjacent minislots (even/odd combinations in Dynamic Segment) on Channel B for certain communication cycles. Which channel handles the Dynamic Segment or Static Segment TX message buffers (MBs) is controlled by the Channel Assignment bits (CHA, CHB) of the Message Buffer Cycle Counter Filter Register (FR_MBCCFRn). The internal Static Segment boundary indicator actually only uses the Channel A slot counter to identify the Static Segment boundary even if the module configures the Static Segment to Channel B (FR_MBCCFRn[CHA]=0 and FR_MBCCFRn[CHB]=1). This results in the Buffer Control Unit waiting for a corresponding data acknowledge signal for minislot:N in the Dynamic Segment and misses the required TX frame transmission within the immediate next minislot:N+1. Workaround 1. Configure the FlexRay module in Single Channel mode (FR_MCR[SCM]=1) and enable Channel B (FR_MCR[CHB]=1) and disable Channel A (FR_MCR[CHA]=0). In this mode the internal Channel A behaves as FlexRay Channel B. Note that in this mode only the internal channel A and the FlexRay Port A is used. So externally you must connect to FlexRay Port A. 2. Enable both Channel A and Channel B when in Dual Channel mode (FR_MCR[CHA=1] and FR_MCR[CHB]=1). This will allow all configured TX frames to be transmitted correctly on Channel B.
ENGR00361074	Defect	[LIN] Avoid multiple definition of STATIC macro

Table continues on the next page...

ID	Subtype	Headline and Description
		Mcal.h ========== 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
ENGR00361051	Defect	[GPT] Update definition of STATIC marco Problem detailed description (how to reproduce it): STATIC macro in base module is redefined by static macro. Driver have to be updated to use standard C keyword "static" instead Expected behavior: Use static macro instead STATIC macro Proposed solution (Optional): Replace STATIC macro by static macro in all files
ENGR00357334	NewWork	[GPT] Update include scheme for RTC NewWork Description: Update include schema so that the Gpt_Cfg.h is included via the Gpt_Rtc_Types.h file to the upper layers. Expected behavior: A correct inclusion schema for the Kinetis GPT driver Proposed solution (Optional): Update Rtc includes similar to PIT or STM
ENGR00359945	Defect	[GPT] Correct Gpt_Rtc_GetTimeElapsed function Problem detailed description (how to reproduce it): In Calypso platform, the APIVAL register value do not compare to counter register. A offset register which is supported by hardware, can be updated to compare to counter register. In continuous mode, measure time elapse is wrong because the APIVAL is not updated in interrupt routine. Expected behavior: Gpt_Rtc_GetTimeElapsed function is correct Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Use a variable to get counter value when interrupt occurs, when expried time, this value is used to calculating time elapse in Gpt_Rtc_GetTimeElapsed function by subtract counter value at the measure time and variable above.
ENGR00357874	NewWork	[GPT] Implement dual clock mode features NewWork Description: IPV_RTC can do exactly in dual clock mode when use set clock mode function: Setup RTC_Prescaler or RTC_Prescaler_Alternate in EB tresos, this

Table continues on the next page...

ID	Subtype	Headline and Description
		parameters will be used to dived clock source which the tests case set clock mode in Normal or sleep Expected behavior: Dual clock mode features works correctly. Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update: - Gpt_Rtc.c - Gpt_Rtc.h to add dual clock mode for IPV_RTC
ENGR00358434	Defect	[GPT] RTC_API timer period is not correct Problem detailed description (how to reproduce it): Configure the GPT channel as attached xdm. With following scenario, the period is changed after each expiration Gpt_Init(&GptChannelConfigSet_0); Gpt_EnableNotification(0); Gpt_StartTimer(0,0x10); Observed behavior: Period is not correct Expected behavior: Period is correct. Proposed solution (Optional): In the Gpt_Rtc_ProcessInterrupt, the compare value is updated as below: RTC_API_SET_COMPARE(Gpt_Rtc_u32TargetValue + u32CounterValue); There is no need to update the compare value in the interrupt routine. It shall only update by Gpt_StartTimer. Remove following code from Gpt_Rtc_ProcessInterrupt /* Get value counter for next interrupt */ /** @violates @ref GPT_RTC_C_REF_3 MISRA 2004 Required Rule 11.1,Cast from unsigned long to pointer */ u32CounterValue = RTC_API_GET_COUNTER(); /* Set new value in compare register */ /** @violates @ref GPT_RTC_C_REF_3 MISRA 2004 Required Rule 11.1,Cast from unsigned long to pointer */ RTC_API_SET_COMPARE(Gpt_Rtc_u32TargetValue + u32CounterValue); Alin Meleandra-B46093: The proposed solution is partially correct it only hold true if the timer is in one-shot-mode but for continuous mode this will not work. The reason for this is that the RTC timer is a Free-running timer so setting a match-compare value this timer will need to take into account the current timer value in order to achieve the expected time-interval.
ENGR00361052	Defect	[GPT] Update definition of STATIC marco Problem detailed description (how to reproduce it): STATIC macro in base module is redefined by static macro. Driver have to be updated to use standard C keyword "static" instead Expected behavior: Use static macro instead STATIC macro Proposed solution (Optional):

Table continues on the next page...

ID	Subtype	Headline and Description
		Replace STATIC macro by static macro in all files
ENGR00361046	Defect	[ICU, DIO, ADC, MCU, PORT] 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
ENGR00357877	Defect	[PORT] Fix error misra Problem detailed description (how to reproduce it): There are 2 error in 2 files on IPV_SIUL2 of PORT module - Reg_eSys_Siul2.h", line 318, MISRA Rule Violated 19.4 (Required): no relevant comment was found for MISRA violation on this line. - Port_Siul2.c", line 749, MISRA Rule Violated 8.10 (Required): no MISRA violation comment was found (maybe wrong format is used).
ENGR00357832	NewWork	[DIO] Fix findings from acceptance review NewWork Description: Please see in attached xls, for each sheet, the column "DIO fix needed". All lines that state a fix is needed in Dio driver development should be addressed. Dio code related findings: - fix code and avoid violation of misra rule 10.3 - fix code and avoid violation of misra rule 10.5
ENGR00361053	Defect	[GPT] Update definition of STATIC marco Problem detailed description (how to reproduce it): STATIC macro in base module is redefined by static macro. Driver have to be updated to use standard C keyword "static" instead Expected behavior: Use static macro instead STATIC macro Proposed solution (Optional): Replace STATIC macro by static macro in all files
ENGR00357817	NewWork	[ICU] Fix Compiler and Misra warnings for Halo

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: Fix Compiler and Misra warnings for Halo 1.0.0 Proposed solution (Optional): For the compiler warnings, there should be another define if channel level prescalers should be used.
ENGR00355980	Defect	[LIN] 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.
ENGR00350896	NewWork	[MCL] Add comments for traceability in Mcl.xdm NewWork Description: Add comments to cover the requirements : CPR-MCAL-732.mcl PR-MCAL-3267.mcl Comments in Mcl.xdm are: MclErrorNotificationDma0_Object MclErrorNotificationDma1_Object MclDisableDemReportErrorStatus_Object
ENGR00361313	NewWork	[MCL] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00361164	NewWork	[MCL] Replace STATIC with static(lowercase)

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: STATIC definition was removed from Base, MCL should use the C keyword static instead.
ENGR00356786	Defect	[MCL] Some DMAMUX sources can not be configured for some channels Problem detailed description (how to reproduce it): -It is not allowed by the configuration schema to configure for the DMAMUX0 channels the source numbers corresponding to the DMAMUX channel numbers configured for DMAMUX1. -It is not allowed by the configuration schema to configure for the DMAMUX1 channels the source numbers corresponding to the DMAMUX channel numbers configured for DMAMUX0. Eg: If there are configured channels 0,1,2 from DMAMUX1(e_DMA17, e_DMA18, e_DMA19), then sources 0,1,2 can not be used by DMAMUX0 channels. The issue is determined by an unique check across all DMA source numbers for a DMAMUX and the replacement of ENABLE attribute with the EDITABLE attribute for the DmaSource0 and DmaSource1 parameters. Observed behavior: -It is not allowed by the configuration schema to configure for the DMAMUX0 channels the source numbers corresponding to the DMAMUX channel numbers configured for DMAMUX1. -It is not allowed by the configuration schema to configure for the DMAMUX1 channels the source numbers corresponding to the DMAMUX channel numbers configured for DMAMUX0. Expected behavior: -It should be possible to configure all dmamux0 sources for dmamux0 channels as long as they are used only for 1 channel in a configuration -It should be possible to configure all dmamux1 sources for dmamux1 channels as long as they are used only for 1 channel in a configuration
ENGR00355982	Defect	[MCL] 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.
ENGR00357763	NewWork	[MCL] Fixed misra issues occurs on Halo RTM1.0.0 NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Fix MISRA issue
ENGR00359661	NewWork	[MCL] Update code to support platforms have different number of channels NewWork Description: On some platforms that have 2 DmaInstance (0 and 1), the number of channels corresponding to DmaInstance is different(For ex: number of channels for DmaInstance0 = 64 channels, DmaInstance1 = 32 channels). But in source code of IPV_DMA just support for number of channels is the same. Update the code to add support for this case.
ENGR00360994	Defect	[MCU] Avoid multiple definition of STATIC macro Mcal.h ========== 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
ENGR00361623	Defect	[MCU] McuFastTransition can't be enabled for STANDBY0 mode Problem detailed description (how to reproduce it): McuFastTransition can't be enabled for STANDBY0 mode
ENGR00355553	Defect	[MCU] Missing default for some variables in the plugin and unnecessary quotes in node:exists() Problem detailed description (how to reproduce it): Customer: 1. In Mcu's module definition, symbolicNameValue is not set for most of the parameters, even though it is a mandatory property. Moreover, McuRamSectionSizeLinkerSym and McuRamSectionBaseAddrLinkerSym has no EcucStringParamDefVariant, which is also mandatory. 2. Wrong argument type of node:exist() function There are a few places where the node:exists XPath function is used with single quotes, like this: node:exists('McuGeneralConfiguration/McuCalloutBeforePerformReset'). The problem with this is that node:exists accepts Node objects, not Strings (according to the documentation of EB tresos Studio 12). Please remove the single quotes. Proposed solution (Optional): 1. symbolicNameValue is missing from some paramters but according to AUTOSAR_TPS_ECUConfiguration.pdf page 86 - If the attribute is not present it shall be assumed to be set to false. McuRamSectionSizeLinkerSym and McuRamSectionBaseAddrLinkerSym are missing <a:da name="DEFAULT" value=""/> and must be added. 2. The single quotes must be removed from the node:exists call

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00358317	NewWork	[MCU] Review and implement changes according to Reference Manual for Calypso 6M/3M RTM1.0.1 Description: RM review: Calypso 6M Hardware documentation MPC5748G_RM_Rev4_DraftB http://compass.freescale.net/go/227152351 Calypso 3M Hardware documentation MPC5746C Reference Manual, Rev. 2, 3/2015 http://compass.freescale.net/go/231236725
ENGR00361296	NewWork	[PORT] Correct the implementation class of PublishedInformation nodes in module Xdm NewWork Description: The definition of nodes included in CommonPublishedInformation container in the xdm configuration files of some modules is incorrect. The modules affected are : ADC, DIO, ETH, FEE, FR, GPT, LIN, PORT. The manifestation of the issue is that Epd tests will fail at generate if the configuration epc file is not updated with the latest software version numbers(SwMajorVersion, SwMinorVersion, SwPatchVersion nodes). Normally, software version numbers will be updated automatically in the output epc file with the M4 mechanism. For all the sub containers in CommonPublishedInformation container (ArReleaseMajorVersion, ArReleaseMinorVersion, ArReleaseRevisionVersion, ModuleId, SwMajorVersion, SwMinorVersion, SwPatchVersion, VendorApiInfix, VendorId) the correct implementation config class is PublishedInformation, not PreCompile/PostBuild. Relevant for this is ASR requirement BSW00402 and also chapter 4.5.2.1 in Tresos documentation - Studio_documentation_users_guide.pdf. The solution for this issue is to chanhe the implementation config clss field of all these nodes from : <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PostBuild">VariantPostBuild</icc:v> <icc:v class="PreCompile">VariantPreCompile</icc:v> </a:a> to <a:a name="IMPLEMENTATIONCONFIGCLASS" type="IMPLEMENTATIONCONFIGCLASS"> <icc:v class="PublishedInformation">VariantPostBuild</icc:v> <icc:v class="PublishedInformation">VariantPreCompile</icc:v> </a:a> 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
ENGR00361315	NewWork	[PORT] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00355635	NewWork	[PORT] Keep the pin level for all GPIO output pins in Port_SetPinMode by a general setting parameter Initial Description: The customer uses Dio_WriteChannel and Port_SetPinMode functions to change the port functional at runtime. 1) Dio_WriteChannel (ch, level) 2) Port_SetPinMode(portpin, mode) MCAL Rainier RTM1.0.1 offers PORT_PIN_LEVEL_NOTCHANGED however when setting the pin with this option, the customer cannot configure the output for GPIO when calling Port_Init. The customer request a parameter in Port General setting so that they can disable Port level setting in Port_SetPinMode. The reason is that with PORT_PIN_LEVEL_NOTCHANGED the customer must configure the port output for many channels. And with the Port General setting parameter for the different projects they can easily select at the configuration phase. e.g: Adding a non-ASR parameter to Port General Setting. UnConfigure Level in Port_SetPinMode PortUnConfigureLevel then define a Macro in Port_Cfg.h if PortUnConfigureLevel = True #define PORT_UNCONFIG_LEVEL STD_ON else #define PORT_UNCONFIG_LEVEL STD_OFF endif In the Port_Ipw_SetPinMode, replace if(PORT_PIN_IN != ePadDirection) { Port_Siul2_SetGpioPadOutput(PinIndex, pConfigPtr); } by #if PORT_UNCONFIG_LEVEL == STD_OFF if(PORT_PIN_IN != ePadDirection) { Port_Siul2_SetGpioPadOutput(PinIndex, pConfigPtr); } #endif
ENGR00358318	NewWork	[PORT] Review and implement changes according to Reference Manual for

Table continues on the next page...

ID	Subtype	Headline and Description
		Calypso 6M/3M RTM1.0.1 Description: RM review: Calypso 6M Hardware documentation MPC5748G_RM_Rev4_DraftB http://compass.freescale.net/go/227152351 Calypso 3M Hardware documentation MPC5746C Reference Manual, Rev. 2, 3/2015 http://compass.freescale.net/go/231236725
ENGR00355967	Defect	[PORT] The short name of Std_Types is Std but not STD_TYPES Problem detailed description (how to reproduce it): According to AUTOSAR_TR_BSWModuleList document, the short name(API service prefix) of Std_Types.h is 'Std', which leads to STD_AR_RELEASE_MAJOR_VERSION being defined instead STD_TYPES_AR_RELEASE_MAJOR_VERSION. Need to replace STD_TYPES_AR_RELEASE_MAJOR_VERSION with STD_AR_RELEASE_MAJOR_VERSION in all places where a check on the version of Std_Types.h file is happening.
ENGR00360466	Defect	[PORT] Update the description of the variables PortPinWithReadback and PortPinDirection in Port_Template.xdm Please update description of the PortPinWithReadback in Port_Template.xdm to: 'Enables/Disables the read back possibility for this pin. Checked box means the Read Back is enabled. When ReadBack is enabled, the Input Bufer of the pin gets enabled by setting the IBE bit in the MSCR (PCR) of the pin. Some alternate functions working as inputs might require having the IBE set to 1, so check this box in order to achieve this.' Please update description of the PortPinDirection in the PortPin container (not the one in the NotUsedPortPin container) in Port_Template.xdm to: 'Selects the direction of the pin (IN, OUT or both) that will be configured by Port_Init() function if the pin is configured as GPIO. If the direction is not changeable, the value configured here is fixed. For the Alternative Function modes (PortPinMode is different than GPIO), the setting in this enumeration control is kept in the port configuration structure and it is used when Port_SetPinMode() is called at runtime to change the mode of the pin to GPIO. If your Alternative Function is an input functionality that requires the IBE bit to be set in the MSCR, please select the checkbox 'PortPinWithReadback'.'
ENGR00361088	Defect	[PWM] 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.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00361316	NewWork	[PWM] Create symbolic link to common reference_list.xml file in buildenv NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00361802	Defect	[PWM] Fix default value MasterModeSelect in xdm file Defaul value of parameter MaterModeSelect is MASTER_MODE_UP_COUNTER. It is not in range MASTER_MODE_UP_BUFFERED_COUNTER, MASTER_MODE_UP_DOWN_BUFFERD_COUNTER Resolution: Change default value -> MASTER_MODE_UP_BUFFERED_COUNTER
ENGR00350954	NewWork	[PWM] High Level header files should not be include for IPV layer directly NewWork Description: The concept of layered driver architecture forbids including of High Level files directly from IPV layer. The only files that can be include are the Wrapper files and Configuration Header file.
ENGR00357919	Defect	[PWM] MCAL / some files exist doubled eMios_Common_Types.h (Gpt, Pwm, Mcl) Siul2_IPVersion.h (Port, Dio) Reg_eSys_eMios.h (Pwm, Mcl) Customer complains that these files are delivered twice within MCAL delivery package (see module short names in brackets). Please remove doubled files which are not intended. This causes confusion on Customer side (in best case) and problems (in the worst case)
ENGR00355984	Defect	[PWM] 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

Table continues on the next page...

ID	Subtype	Headline and Description
		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.
ENGR00354213	NewWork	[PWM]Full feature coverage for eMIOS platforms NewWork Description: eMIOS platforms do not support the following AUTOSAR features: - selection of parameters PwmPeriodUpdatedEndperiod and PwmDutycycleUpdatedEndperiod is not possible, These parameters are set to always "true". These parameters can be made available in configuration by selection the correct operation mode. (i.e. for OPWFM it is possible to select either immediate update or next period update by the MODE bits) - OPWM mode is not supported (for Calypso) - CAOPWMDTI (Center Aligned Output Pulse Width Modulation with Dead Time Insertion) is not supported - Some Operation modes are not are not supported for all eMIOS platforms. (i.e. OPWM) GMRT requests that all feature should be supported by our drivers.
ENGR00355418	Defect	[SPI] Remove the requirement "PR-MCAL-3225.spi" implementation in the code Problem detailed description (how to reproduce it): Calypso chip not support TSB mode, so the implementation for requirement "PR-MCAL-3225.spi" should be removed
ENGR00358320	NewWork	[SPI] Review and implement changes according to Reference Manual for Calypso 6M/3M RTM1.0.1 Description: RM review: Calypso 6M Hardware documentation MPC5748G_RM_Rev4_DraftB http://compass.freescale.net/go/227152351 Calypso 3M Hardware documentation MPC5746C Reference Manual, Rev. 2, 3/2015 http://compass.freescale.net/go/231236725
ENGR00357347	Defect	[SPI] The default mode for SPI with delivered level 2 The default mode when SPI configured with delivered level 2 is POLLING mode. However, this information is not stated in the integration manual. From my point of view, the default mode should be INTERRUPT mode rather than POLLING mode. or please update the UM to reflect the current behavior. From that wording in 3.6.2.1 Enumeration Spi_AsyncModeType, It may be still

Table continues on the next page...

ID	Subtype	Headline and Description
		difficult for customers to conclude that the SPI mode is POLLING and they need to call Spi_SetAsyncMode(SPI_INTERRUPT_MODE) or Spi_SetHwAsyncMode(hw, SPI_INTERRUPT_MODE) to switch into INTERRUPT mode if using the interrupt. Actually, I got the same question from different customers so I have inputted this CR.
ENGR00355985	Defect	[SPI] 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.
ENGR00355569	Defect	[SPI] Wrong comment block usage Problem detailed description (how to reproduce it): Customer: In Spi_Cfg.c, Spi_PBcfg.c, Spi_Lcfg.c: The comment block at the beginning of each file shall be a C comment, not a template comment, in order to get it written to the generated files
ENGR00355986	Defect	[WDGIF] 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.
ENGR00361319	NewWork	[WDG] Create symbolic link to common reference_list.xml file in buildenv

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: Create symbolic link to common reference_list.xml file in buildenv. All modules need to use a table defined by this xml in their UM, IM, TS.The content of this file is usually the same across drivers and across documents. An instance of this file has been created in a central location (buildenv) here: SASW\JSW_MCAL\XPC56xx\dev\test\test\doc\docgen\reference_list.xml. Module documentation can use this version directly via symlink (just like variables_common.xml), this will eliminate the need of updating this file for each module in the future. If your module requires a non-standard documentation list, you should not use the common file. Please check the content of your original file before using this.
ENGR00361245	NewWork	[WDG] Replace STATIC with static(lowercase) NewWork Description: STATIC definition was removed from Base, WDG should use the C keyword static instead.
ENGR00355987	Defect	[WDG] 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

4.6 RTM 1.0.0

ID	Subtype	Headline and Description
ENGR00328011	NewWork	[ADC] Implement new cPRD requirements related to CTU control mode. Implement the following New cPRD requirements: CPR-MCAL-726 The Adc driver shall provide configuration support and optional functions for enabling and disabling CTU control mode for an ADC unit. These optional APIs shall be called Adc_EnableCtuControlMode(Adc_UnitType Unit) and Adc_DisableCtuControlMode(Adc_UnitType Unit). When a unit works in CTU control mode, no other conversions shall run in parallel(Adc). The only conversions occurring shall be the ones defined in the CTU configuration. CPR-MCAL-727 The service Adc_EnableCtuControlMode shall report error to DET (if enabled) when:

Table continues on the next page...

ID	Subtype	Headline and Description
		- Adc_Init() was not called before this function. In this case ADC_E_UNINIT shall be reported. - Parameter Unit is not configured or a CTU unit is not configured for it . In this case ADC_E_PARAM_UNIT shall be reported. - Hw unit status is busy (any Injected, Hw triggered or SW triggered conversions are ongoing) - In this case ADC_E_BUSY shall be reported. - Ctu control mode is already enabled for this unit. In this case ADC_E_BUSY shall reported. CPR-MCAL-728 The service Adc_DisableCtuControlMode shall report error to DET (if enabled) when: - Adc_Init() was not called before this function. In this case ADC_E_UNINIT shall reported. - Parameter Unit is not configured or a CTU unit is not configured for it. In this case ADC_E_PARAM_UNIT shall be reported. - Ctu control mode is not already enabled for this unit. In this case the ADC_E_IDLE shall be reported.
ENGR00352154	Defect	[ADC] Adc Prescaler is not configurable Problem detailed description (how to reproduce it): In the ADC configuration it is not possible to set the prescaler of an Adc to something different than 2 (parameter is not writable). In addition to that, the code uses the macro divide by one instead of divide by two. This is in contradiction with the fact that you cannot configure the clock divided by one Observed behavior: AdcPrescale is readonly (2) Expected behavior: AdcPrescale is configurable Proposed solution (Optional): In Xdm file, Change AdcPrescale to configurable. In template file (Adc_Cfg.c, Adc_PBAdc.c) , generate the ADC_CLOCK_PRESCALER_DIV1_U32 or ADC_CLOCK_PRESCALER_DIV2_U32 for ADCLKSEL bit depending on the customer configuration.
ENGR00327216	Defect	[ADC] AdcSetOnceRegisters functionality is not correct if more than one ADC unit is used Problem detailed description (how to reproduce it): If AdcSetOnceRegisters option is set and more than one ADC unit is used, the conversion time and presanmplin registers are not configured correclty - instead of configuring the values for each unit, all units will be configured with the values for the last unit. 1) Hw unit0 consists of: Group0: Id -> 0 Group1: Id -> 2 2) Hw unit1 consists of: Group2: Id -> 1. Note: The Id of group 2 must not be the maximum. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] 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): [...]
ENGR00335838	NewWork	[ADC] Add Dem inclusion afrer it already removed from Base module NewWork Description: Add Dem inclusion afrer it already removed from Base module Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00330127	NewWork	[ADC] Architectural improvements - move generic processing to high level driver NewWork Description: ADC module File Structure - Adc.h - shall not be imported by the IP layer IP layer - Separate ADC logic from IP logic - a lot of logic can be placed in the upper layer - helps at mantainability and porting Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00352774	NewWork	[ADC] Clarifications for all available hardware triggering features in the User Manual NewWork Description: I think we need to include some clarifications for all available hardware triggering features in the User Manual, create a sub-chapter of "Driver Design Summary". There should be a list there of all possibilities for HW triggering, how to configure them and what are their special features. Currently, it is not very visible what triggering can be done with/without BCTU, what it means to have BCTU control mode, MHT, single mode, list mode. All

Table continues on the next page...

ID	Subtype	Headline and Description
		these are just mentioned briefly in the hardware RM or the requirements, but some of them are cool features and deserve to be presented in a nicer way, otherwise the customer might never know they are there. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00274628	NewWork	[ADC] DMA initiates transfer before receiving trigger from ADC after ADC group configured SW trigger, Continuous uses DMA Initial Description: When a group that uses DMA transfer is running in continuous mode , it sends triggers to the DMA after every conversion of the channels it contains. After the last valid round of data is transferred (depending on the nr of samples in the group), DMA triggering in DMA is stopped, ADC conversions are stopped and DMA hardware request is stopped ? as would be expected. However, when the next group starts that uses DMA (in this case Group_3), the DMA transfer complete ISR occurs before the transfer is triggered by the ADC ? immediately after the DMA hardware request is re-enabled. For HW triggered ADC groups, it occurs before the eTimer is started. the issue can be reproduced for SW triggered groups ? the ISR is triggered before I set the NSTART bit. Note that this is visible in the test only if some channels of the group that runs after the continuous group have never been converted (as is the case here, with Group_3 and the current configuration). If all channels have been converted before, old data will be read and the problem will be transparent. (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:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
ENGR00346709	NewWork	[ADC] Empty Note field in the External Assumptions IM chapter NewWork Description: Some of the drivers have in their Integration Manuals - Chapter: External Assumptions, notes stating irrelevant information, e.g.: - 'Ok' (for ADC) / - 'SM', 'No able to check it.' (for LIN) - 'SM' (for SPI) Expected behavior: The irrelevant comments for IM's Notes field shall be updated in DOORS module and regenerated into IMs.
ENGR00336415	Defect	[ADC] Fix Misra errors Problem detailed description (how to reproduce it): After the latest code refactoring, we have 2 more Misra errros that neeed to be fixed: - a 8.10 rule violation (8.10 (req): %s: All declarations and definitions of objects or functions at file scope shall have internal linkage unless external linkage is required.) for a function that is used from 2 source files and needs external linkage - a 1.4 rule violation (1.4 (req-): The compiler/linker shall be checked to ensure that 31 character significance and case sensitivity are supported for external identifiers.) for the usage of DISABLE_MCAL_INTERMODULE_ASR_CHECK identifier. 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): [...]
ENGR00340951	NewWork	[ADC] Fix Misra errors NewWork Description: fix / add comments for Misra errors in ADC files 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): [...]
ENGR00349441	NewWork	[ADC] Fix code checklist review findings - Part 3 NewWork Description: Fix code after review the code against code checklist. The reviewing has been done by ENGR00348720 See the attachment for code review report [...] Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00331140	NewWork	[ADC] Implement requireemnt CPR-MCAL-730 in SMCAL NewWork Classification: (internal task, improvement, feature request) This is Requirement PR-MCAL-3232: The Adc driver shall provide an optional function, API and configuration parameters to check if an injected conversion (HW/SW) is ongoing. This optional API shall be called "Adc_GetInjectedConversionStatus(ADC_HW_UNIT)" Per default this optional functionality, API and configuration parameters shall be disabled. For SW injected conversion the function checks the field SwInjQueueIndex of the Adc_UnitStatusType structure. For HW injected conversions (with more than one channel) the function can use both HwQueue and HwQueueIndex fields of the Adc_UnitStatusType structure. This function returns HW unit status as ADC_BUSY, 1) If at least one software triggered injected conversion group on the same HW unit is in progress or 2) If any hardware triggered group conversion on the same HW unit with more than one channel is in progress. Limitation: User should ensure that the HW unit is in ADC_IDLE status by calling Adc_GetInjectedConversionStatus(), before calling "Adc_StartGroupConversion/Adc_EnableHardwareGroup" for new injected conversion group. After discussing with Frantisek, it was revealed that the purpose of this requirement was initially to poll the status of injected conversions in the driver. At the time this was created, HW Triggered coversions cound convert only one channel, and for the rest of the group's channel an internal injected conversion was created. this is not the case anymore in SMCAL codebase - now, injected and HW triggered conversion share the mask registers JCMRx adn the interrupt sources. In conclusion, the text of PR-MCAL-3232 should be updated to remove the "with more than one channel" - this would be the correct definition for this API for SMCAL. I attached the email discussion I had with Frantisek about this requirement.

Table continues on the next page...

ID	Subtype	Headline and Description
		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...)
ENGR00352738	NewWork	[ADC] Improvement for Bctu configuration NewWork Description: There are some improvement should be implemented for BCTU configuration: - User will select hardware channel for single mode by a text box (It's friendly than put a number as before). - User no need to configure LADDR value in LIST mode. This value will be calculated automatically by template files (Adc_Bctu_Cfg.c and Adc_Bctu_PBcfg.c). - Not allow user configure LIST mode with only one channel. If only one channel, recommend user to use SINGLE mode. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00352402	Defect	[ADC] Parameters using ENABLED attribute are removed from the data model Problem detailed description (how to reproduce it): A lot of parameters (spread over all modules) that are visible within Tresos GUI, but greyed out as not editable or conditionally visible (depending on different configuration parameters) are missing within the exported AUTOSAR configuration file (see attached Tresos_GUI.jpg). After import of this configuration file the configuration validation fails, because those parameters are mandatory (multiplicity 1 .. x) within the module description file in AUTOSAR format. The export has to consider those parameters. What generates this problem is that when Tresos generates the EPD, it removes all nodes which have the attribute ENABLE=False. Proposed solution: To avoid Tresos removing AutoSAR and Vendor Specific mandatory parameters (multiplicity 1 .. x) when epd file is generated, in xdm file, their attribute ENABLE (if it exists) shall be replaced by EDITABLE attribute. The multiplicity can be observed in the .epd files in the plugin's autosar folder. So, the solution is the replacement of: <a:da name="ENABLE" type="XPath"> with <a:da name="EDITABLE" type="XPath">

Table continues on the next page...

ID	Subtype	Headline and Description
		This shall be done for all AutoSAR mandatory and Vendor specific parameters.
ENGR00330718	NewWork	[ADC] Remove DMA trigger source from the trigger list NewWork Description: DMA trigger source should be removed from the trigger list Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00330697	NewWork	[ADC] Remove all production error reporting to SERR and use only DEM NewWork Description: All production error reporting shall be done using the AutoSar defined mechanism - DEM. All calls to SerrNotifyError shall be replace by DemReportErrorEvent. Proposal for implementation of DEM reporting is in the attachment. For details on how to implement Dem handling please see the attachement Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00332152	NewWork	[ADC] Review and update User manual and Integration Manual NewWork Description: Please review in all manuals the following items: - Interrupt service routine and interrupt vector are documented correctly. - The non-autosar/ Freescale specific parameters are documented with a correct description. - Remove unavailable parameters/APIs in the platform. - Deviation from ASR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00345153	NewWork	[ADC] Separate code for Software Injected conversions with precompile defines for code optimization

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: Separate code for Software Injected conversions with precompile defines for code optimization Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00349658	Defect	[ADC] Software trigger group should not be allowed to configure if Adc_StartStopGroup API is switched off. Problem detailed description (how to reproduce it): The driver should raise an error at the configuration time when configure a software trigger group with Adc_StartStopGroup API is switched off. 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): [...]
ENGR00334413	NewWork	[ADC] Support Normal conversion with Hardware trigger and Software Injected trigger concurrent request in the same hardware unit Hi, This is for Matterhorn SMCAL. Currently, the hardware trigger group always uses hardware injected trigger. However, the customer project needs two groups which are configured in Hardware trigger and software injected trigger as below. Adc_EnableHardwareTrigger(G0); Adc_StartGroupConversion(G1); The group G0 is triggered periodicaly by Hardware (ATOM/TOM) then conversion data is transfered by DMA. This use-case is not supported then the DET error is reported in Adc_StartGroupConversion(G1). Can we support this use-case? From reference manual, this could be done by enabling conversion type (Normal conversion/Injected Conversion similar to the software trigger group) for hardware trigger group. If ADC_CONV_TYPE_INJECTED is chosen, the group will be configured as current implementation, if ADC_CONV_TYPE_NORMAL is chosen, the group will be configured in the

Table continues on the next page...

ID	Subtype	Headline and Description
		normal conversion mode and triggered by hardware trigger.
ENGR00352461	NewWork	[ADC] Support for HW triggered ADC groups on externally multiplexed (ANX) channels - remove implementation of PR-MCAL-3317 NewWork Classification: (internal task, improvement, feature request) In AR_BOL_BSW_329 (PR-MCAL-3301, PR-MCAL-3317) you have introduced some new parameters for multiplexed inputs acquisition. The idea was to allow to convert ANX channels, with and without stabilization delay (decided at runtime based on the ADC group that was converted). We want to use an ANX channel (with stabilization delay) with a HW triggered ADC group. This is not allowed anymore in MCAL AR3.2, we get an error from the code generator. It should be allowed to use ANX channels in HW triggered ADC groups if the DSDR is not changed at runtime (for example if we want to use stabilization delay for all conversions done on ANX channels). The project that has this problem is using ANX channels with HW and SW triggered ADC groups and for all conversions they need the same stabilization delay. This should be possible if we set the AdcMuxDelaySupport to TRUE and we set to false the AdcMuxDelaySupportPerGroup. The DSDR should be written at init time and not re-written at runtime. PR-MCAL-3317 does not allow HW triggered groups with externally multiplexed ANX channel and as such needs to be reconsidered. 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...)
ENGR00343698	Defect	[ADC] The code can't build successfully if MHT feature is ON Problem detailed description (how to reproduce it): We have a compiler error and some warning raised if compiles code with MHT feature is ON. See log below: "C:/EB/tresos/1421/plugins/Adc_TS_T2D34M9I2R0/src/Adc.c", line 1177: warning #550-D: variable "NoGroupsInSwNormalQueue" was set but never used VAR(Adc_QueueIndexType, AUTOMATIC) NoGroupsInSwNormalQueue = 0U; ^ "C:/EB/tresos/1421/plugins/Adc_TS_T2D34M9I2R0/src/Adc.c", line 1178: warning #550-D: variable "NoGroupsInSwInjectedQueue" was set but never used VAR(Adc_QueueIndexType, AUTOMATIC) NoGroupsInSwInjectedQueue = 0U; ^ "C:/EB/tresos/1421/plugins/Adc_TS_T2D34M9I2R0/src/Adc.c", line 1181: warning #550-D: variable "GroupType" was set but never used VAR(Adc_GroupConvType, AUTOMATIC) GroupType = ADC_CONV_TYPE_NORMAL;

Table continues on the next page...

ID	Subtype	Headline and Description
		^ "C:/EB/tresos/1421/plugins/Adc_TS_T2D34M9I2R0/src/Adc.c", line 2574: error #20: identifier "pGroupPtr" is undefined Adc_UnitStatus[Unit].HwQueueGroupType = pGroupPtr->IsMHTGroup; Please fix this issue and check for other compiler error and warning issues. 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): [...]
ENGR00341117	NewWork	[ADC] The high threshold value should not allow to configure lower than low value NewWork Description: The driver should raise an error in the configuration time if high threshold value is configured lower than low value or low threshold value is configured greater than high value Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00341423	Defect	[ADC] The injected conversion should be aborted when call Adc_StopGroupConversion Problem detailed description (how to reproduce it): The injected conversion should be aborted when Adc_StopGroupConversion is called. teh injected conversions were ignored by this API because the RM stated that once started, an injected conversion could not be stopped. But the latest RM states that even the injected conversion can be stopped. Preconditions: [...] 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): [...]
ENGR00330341	Defect	[ADC] Update Det error checks for Adc_EnabelCtuControl Problem detailed description (how to reproduce it): Update Det error checks for Adc_EnabelCtuControl: Adc_StartGroupConversion, Adc_EnableHardwareTrigger shoud throuw ADC_E_BUSY if the unit is in CTU control mode. 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): [...]
ENGR00332321	Defect	[ADC] Update IM MCL dependency for ADC, SPI when use in DMA mode and PWM, GPT, ICU Problem detailed description (how to reproduce it): Please add into Chapter 5: Mcl should be init before ADC, SPI in DMA mode. MCL should be called before ICU, PWM, GPT (ATOM used). Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00348662	Defect	[ADC] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the

Table continues on the next page...

ID	Subtype	Headline and Description
		details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described.

Table continues on the next page...

ID	Subtype	Headline and Description
		GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00353150	Defect	[ADC]The function Adc_EnableHardware does not raise DET error when the unit ADC is in CTU control mode The function Adc_EnableHardware does not raise DET error when the unit ADC is in CTU control mode How to reproduce: - Adc_EnableCtuControlMode(1); /* Enable CTU control mode for unit 1 */ - Adc_EnableHardwareTrigger(Group0); -> Expect error ADC_BUSY. Test case: Adc_TC_0188
ENGR00336591	Defect	[BASE] Add comments for Misra errors in MemMap.h Problem detailed description (how to reproduce it): Add comments for Misra errors in MemMap.h stub file 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00339914	Defect	[BASE] All assembly load instructions use VALUE@h instead of VALUE@ha Problem detailed description (how to reproduce it): All assembly load instructions use VALUE@h instead of VALUE@ha to avoid compensating sign extension ? VALUE@h: Only the high 16-bit part (bits 0-15). ? VALUE@ha: Like @h, but adjusted to compensate for sign extension applied by an "addi VALUE@l" on the same register. Proposed solution (Optional): Update MCAL.h and replace @ha by @h
ENGR00351077	NewWork	[BASE] Create quality package for Calypso 6M RTM1.0.0 Description: Create final Test Report (if ITG is not involved in the release) Create final MISRA report (if ITG is not involved in the release) Create final compiler warnings report (if ITG is not involved in the release) Create final VSMD report Generate Traceability Matrix Report (if it is not EAR release) Create UM/IM
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

Table continues on the next page...

ID	Subtype	Headline and Description
		'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: [...] 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.
ENGR00330509	NewWork	[BASE] Remove SOC specific config params for DMA and DMAMUX NewWork Description: Following compiler switches have to be removed from Base, because they are MCL specific and generated usinging MCL resources: #define DMA_NB_CONTROLLERS (0x01U) #define DMA_NB_CHANNELS (0x40U) #define DMAMUX_NB_CHANNELS (DMA_NB_CHANNELS) #define DMAMUX_CTRL_MAX_CHANNELS (0x10U) #define DMAMUX_CTRL_MAX_CHANNELS_MASK (0x0FU) #define DMA_CTRL_MAX_CHANNELS (0x40U) #define DMA_CTRL_MAX_CHANNELS_MASK (0x3FU) #define LIN_USE_DMA_LLD (STD_OFF) #define ADC_USE_DMA_LLD (STD_ON) #define SPI_USE_DMA_LLD (STD_ON) #define MCU_USE_DMA_LLD (STD_ON) #define ICU_USE_DMA_LLD (STD_ON) Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00333328	NewWork	[BASE] Remove all production error reporting to SERR and use only DEM NewWork Description: All production error reporting shall be done using the AutoSar defined mechanism - DEM. All calls to SerrNotifyError shall be replace by DemReportErrorEvent. Proposal for implementation of DEM reporting is in the attachment. For details on how to implement Dem handling please see the attachement

Table continues on the next page...

ID	Subtype	Headline and Description
		Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00330634	NewWork	[Base]Add IPV_DMA version in Soc_Ips.h NewWork Description: Add IPV_DMA version 0x02000011 Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00325961	Defect	[DIO] GPI pin supporting analog functionality cannot be configured as DIO pin Problem detailed description (how to reproduce it): GPI pin that is also supporting analog functionality cannot be configured as DIO pin. Ex.: MSCR[76] According to Reference Manual following modes are supported: GPI[76] ADC1_AN[6]/ADC3_AN[5] But when configuring in DIO (DioPortId = 4, DioChannelId 12) following error is raised: Value "12" of node "/AUTOSAR/TOP-LEVEL- PACKAGES/Dio/ELEMENTS/Dio/DioConfig/DioConfig_0/DioP ort/Port_E/DioChannel/xyz/DioChannelId" is out of range: For input string: "This channel is not available in the selected package.Please see Elt. Desc note for a list of available channels for selected package." Preconditions: Configuration of GPI pin supporting analog functionality. Test Case ID (internal TC that caught the defect) - optional N/A Trigger: N/A Observed behavior: Configuration error (see above) Expected behavior: GPI pin should be configurable in DIO module. 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
ENGR00345231	Defect	[DIO] Port ID missing in user manual Problem detailed description (how to reproduce it): MPC5748G version 0.9.0 In DIO user manual, DioPortID of PORTI and PORTO is missing. Please clarify. 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): [...]
ENGR00344794	Defect	[DIO] fix misra report Problem detailed description (how to reproduce it): There are some error when create report misra for DIO on Rayleigh 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): [...]
ENGR00347134	NewWork	[DIO][PORT]Update register access masks for ARM 64 bit architecture NewWork Description: Update register access masks for ARM 64 bit architecture 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
ENGR00333861	NewWork	[ETH] Analyze data flow anomalies from the StaticAnalysis report NewWork Description: Analyze and fix all problems for the data flow anomalies in the static analysis report. All suspicious variables usage that are not errors need to be documented in the Remarks column. More details on data flow anomalies can be found in the attachement
ENGR00345016	Defect	[ETH] Compiler warnings II Problem detailed description (how to reproduce it): The Eth driver generates following warnings with the Diab compiler (with -Xlint option): ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 313: warning (dcc:1516): parameter u8Ctrl is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 507: warning (dcc:1516): parameter u8Ctrl is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 605: warning (dcc:1516): parameter u8Ctrl is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 618: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 626: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 634: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 642: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 650: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 658: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 666: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 674: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 682: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 690: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 698: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 724: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 736: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 749: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 764: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 774: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 787: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 801: warning (dcc:1516):

Table continues on the next page...

ID	Subtype	Headline and Description
		parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 809: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 817: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 825: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 833: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 841: warning (dcc:1516): parameter u8CtrlIdx is never used ".\Eth_TS_T2D32M10I0R0\src\Eth_LLD.c", line 851: warning (dcc:1516): parameter u8CtrlIdx is never used Customer knows that the warnings are documented in code as implemented by ENGR00313710 but still requests to remove the warnings. Customer also has a solution: (void) (x) in code of function would prevent the warning. That's the way what we are doing in our modules. 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): (void) (u8CtrlIdx); in code of function would prevent the warning.
ENGR00355142	Defect	[ETH] Fix Misra error Problem detailed description (how to reproduce it): There is new misra error due to change of DET. This CR will fix this issue. 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): [...]
ENGR00335960	Defect	[ETH] Fix comments reported by review checklist NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		There are some more findings related to naming convention. However, becasue of time restricted, these findings won't be fixed now and will be done in this follow up CR to prevent any problem. The findings are: - MISRA comments contain LLD in the label name -> for example Eth_ENET_LLD_h_REF_1. Please remove LLD from the name of the MISRA labels - All variables which are CONST have co prefix in their name. There is no rule in code Checklist which says that CONST var should have the prefix "co" - Some of the defines are still containing LLD, please remove this from the defines name -> for example ENET_LLD_TXB_LOCK -Some defines does not have Suffix according to checklist (U8, U32 etc). In Eth_ENET_LLD.h, Eth_ENET_Counters.h -> for example ENET_LLD_TXB_LOCK, ENET_CTR_IEEE_R_DROP, ENET_TCSR_TF_W1C, ENET_EIR_BABR_W1C Code checklist available at (also attached): http://compass.freescale.net/livelink/livelink?func=ll&objId=224108405&objAction=browse Tip: To help automatize checking of some coding rules, you can use AUTO_TOOLS Review Assist tool (BLN_REVIEW_ASSIST_01.00.00). But please be aware this tool is not yet qualified, so all the rules need to be checked by human review.
ENGR00344839	Defect	[ETH] Fix misra error Problem detailed description (how to reproduce it): There are some misra errors. This CR will fix or comment these errors. 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): [...]
ENGR00337677	NewWork	[ETH] Rename file for IPW and IPV layer NewWork Description: There isn't rule for naming of files in IPW and IPV. However, we'd better to rename to have common naming in all ASR driver. So Eth_LLD will rename to Eth_IPW and Eth_ENET_LLD will be Eth_Enet. Expected behavior: New naming convention 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): Rename file in CC.
ENGR00348677	Defect	[ETH] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC

Table continues on the next page...

ID	Subtype	Headline and Description
		FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_3
ENGR00326876	NewWork	[FEE] Add information regarding FEE virtual page size also to the IM Original Headline: [FEE] FEE page size smaller 32 bytes Initial Description: We need to clarify if a page size smaller then 32bytes (customer has currently 8) is not a robustness issue due to the fact, that the flash cache lines have a size of 32 bytes. To be checked if an ECC error in any of the 32 bytes will affect the complete cache line and might cause collateral damage to sibling blocks. (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: [...] Reported release baseline: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?)

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] Expected behavior: [...] 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...)
ENGR00333862	NewWork	[FEE] Analyze data flow anomalies from the StaticAnalysis report NewWork Description: Analyze and fix all problems for the data flow anomalies in the static analysis report. All suspicious variables usage that are not errors need to be documented in the Remarks column. More details on data flow anomalies can be found in the attachement
ENGR00341748	Defect	[FEE] Change default value of FeeNvmJobEndNotification and FeeNvmJobErrorNotification Problem detailed description (how to reproduce it): The default value of FeeNvmJobEndNotification and FeeNvmJobErrorNotification should be NvM_JobEndNotification/NvM_JobErrorNotification respectively
ENGR00338345	Defect	[FEE] FEE blocks lost after cluster swap if there is INCONSISTENT block Problem detailed description (how to reproduce it): MPC564XC_MCAL4_0_RTM_1_0_1 Customer reported an issue where secret keys stored by FEE driver are lost. See attached data dump of DFLASH and NVM,FEE,FLS configuration. After investigation we found out that all blocks after INCONSISTENT one are lost after cluster swap. In customer project FeeBlockAlwaysAvailable parameter is configured to false. How to reproduce: 1. Write block with FEE block number 56 (0x38) 2. Before data are written perform reset or power down (i.e. only block header is written) 3. Application starts-up and after FEE initialization the block 56 (0x38) is correctly detected as INCONSISTENT 4. Write another blocks until cluster swap is done 5. After cluster swap dataAddr of all blocks with block number higher than 56 is wrong (shifted by size of block 56)

Table continues on the next page...

ID	Subtype	Headline and Description
		6. When reading data of blocks with block number higher than 56 wrong data are returned 7. The issue seems to be caused by Fee_JobIntSwapClrVldDone function which updates the dataAddrIt by block size of inconsistent block (0x220), but as there are no data stored for this INCONSISTENT block this results in wrong dataAddrIt for all subsequent blocks In attached DFLASH dump, and in attached xlsx file with parsed data of this dump, it can be seen that data of all blocks are correctly stored in clusterA (0x800000), but wrongly stored in other three clusters. After cluster swap from clusterA (0x800000) to clusterC (0x808000) all blocks with number > 56 have wrong data even if they are still VALID. This issue is related to a change done in version 1.0.1 by ENGR00258972 "[FEE] Swap still not working if data is corrupted for ageing of Flash". See attached code comparison of Fee_JobIntSwapClrVldDone code comparing 1.0.0 and 1.0.1 version (fee_inconsistent_dataaddr_moved.png). In version 1.0.1 the dataAddrIt is updated even for INCONSISTENT block which was not done in version 1.0.0. Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00322451	NewWork	[FEE] Improved support of 'Immediate Data' NewWork Classification: (internal task, improvement, feature request) improvement Expected behavior: avoid cluster swaps in case of immediate data write requests Requirement source: after discussion with customer these requirements shall apply: Req1 For each cluster group, the FEE module configuration shall calculate the size of a "reserved space". The "reserved space" is an area of such a size that all configured immediate blocks including needed overhead fit into it exactly once. Req2: The Fls shall trigger a cluster swap in the following cases: ? If the execution of a write operation to a non-immediate block would touch the reserved space in the active cluster. ? If Fee_EraseImmediateBlock() was called for an immediate block which has been already written into the reserved space. ? If an immediate block was attempted to be written which already has been written into the reserved space. Its write attempt will then be performed in the new cluster. Req3: The Fee_EraseImmediateBlock() function shall not physically pre-allocate the block by writing its header and reserving data space. Req4: The functionality of the Fee_EraseImmediateBlock() shall be pre-compile time configurable for either a legacy or normal mode. In legacy mode, the Fee_EraseImmediateBlock() function shall invalidate the

Table continues on the next page...

ID	Subtype	Headline and Description
		specified immediate block only then if it has been written (is not empty). In normal mode the status of the specified immediate block shall not be touched at all.
ENGR00332744	Defect	[FEE] Incorrect description of status flags in UM, ch. 4.1 Problem detailed description (how to reproduce it): Valid and invalid flags (chapter 4.1) are improperly described in the UM. Observed behavior: Flags are described as uint32 typed with values of 0x81 and 0x18 respectively. Reported release baseline: BLN_FEE_MCAL_3.0_XPC56XXK_01.02.00 When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: Flags should be descibed as uint8 value and the blank padding. Proposed solution (Optional): Fix the manual.
ENGR00331837	Defect	[FEE] Misleading wording regarding FEE callbacks in the IM Problem detailed description (how to reproduce it): The paragraph ? Fee_Job(End/Error)Notification: This callback notification is used by the underlying FLS module to report the successful/failed end of an asynchronous operation. may lead to a wrong assumption these callbacks are necessary in FLS synchronous mode only. Observed behavior: User didn't configure the notifications in FLS properly. Expected behavior: Unambiguous formulation. Proposed solution (Optional): ? Fee_JobEndNotification: This callback notification is used by the underlying FLS module to report the successful end of a FLS operation. ? Fee_JobErrorNotification: This callback notification is used by the underlying FLS module to report the failure of a FLS operation. Both callbacks must be configured in the FLS module (notifications) regardless of FLS mode (synchronous or asynchronous).
ENGR00330097	NewWork	[FEE] Reflect "Optimized ECC handling in code flash segments" in FEE manuals NewWork Description: Update "FEE Virtual Page Size"-related information in FEE IM and UM according to the optimized ECC handling implemented in FLS. 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): [...]
ENGR00337516	NewWork	[FEE] Remove CER mechanism from sMCAL code base. As per iTWG decision the CER mechanism need to be removed from sMCAL codebase
ENGR00327995	NewWork	[FEE] Unify signedness of module/ASR version defines NewWork Description: Some defines related to intra- and intermodule version checks are signed, other unsigned. As a result, MISRA violation has been raised (and commented). If all relevant defines are of the same type, no issues would be encountered. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Fix the type of the defines. Remove also unnecessary global MISRA comment definition.
ENGR00341783	Defect	[FEE] Update Fee_Init description in UM Problem detailed description (how to reproduce it): Fee_Init is Asynchronous according ASR spec FEE085 but it is documented as Synchronous. Please correct the UM.
ENGR00341646	NewWork	[FEE] Update User manual about limitation on block size NewWork Description: The customer configures two sectors (16KB) per cluster then add a 20KB block. At generation time, the error message occurs. The limitation: The cluster must be able to hold one instance of each block and additionally one instance of any block (which translates into all plus largest in less precise wording). And the management overhead must be also taken into account (i.e. two exactly-16KiB-block instances won't fit into a 32KiB cluster). Please update the limitation in driver limitation.
ENGR00342995	Defect	[FLS] Change default value of FlsJobEndNotificationand FlsJobErrorNotification Problem detailed description (how to reproduce it): The default value of FlsJobEndNotificationand FlsJobErrorNotification should be Fee_JobEndNotification/Fee_JobErrorNotification respectively

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00352189	Defect	[FLS] FLS driver doesn't give access to all flash blocks Problem detailed description (how to reproduce it): The customer wants to implement a bootloader using FLS driver. But FLS driver doesn't give access to some areas of flash (for instance the 16kb block at x00F8C000). Is it possible to include all flash sectors for the next release? 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): [...]
ENGR00338751	Defect	[FLS] FLS user manual incomplete A customer noticed that the user manual of the FLS driver in MCAL 4.0 Beta 0.9.0 is incomplete and incorrect: table 5-1 with the list of sectors doesn't match the list available in the configuration. Some sectors appear missing, and the sizes are not correct.
ENGR00330847	NewWork	[FLS] FMEA reports shall contain a link between safety measures and requirement IDs All FMEA shall contain the link between the safety measure and the requirement ID or the external assumption generated due to that safety measure. A new column shall be added in our current FMEAs to indicate what is the DOORS ID and the text of the requirement. For more details see the attachement..
ENGR00355374	NewWork	[FLS] Fix misra error 16.10 for Det_ReportError NewWork Description: Returned value for Det_ReportError function is not read. As a result, MISRA error is raised for Rule 16.10 ("If a function returns error information, then that error information shall be tested") Proposed solution: Comment MISRA error. Det_ReportError function always return E_OK error code, so there is no reason to test it.
ENGR00346698	Defect	[FLS] Fls_LoadAc and Fls_UnloadAc writing exceed Fls_ACWriteSize by 4 bytes

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): The customer reported following defect in SMCAL. MCAL 0.9.0 HF2 MPC574xG, FLS, file Fls.c, functions Fls_LoadAc and Fls_UnloadAc. The end condition in the for loop is for(; romPtr <= romEndPtr; romPtr++) 4 extra bytes are written It should be: for(; romPtr < romEndPtr; romPtr++) 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): [...]
ENGR00339663	NewWork	[FLS] Implement new prefetch buffer invalidation approach NewWork Description: Generic stuf: Fls.xdm - remove the whole variable "FlsDisableAccessToFlashControllerRegisters" Fls_Cfg.h - remove macro FLS_DISABLE_PFLASH_ACCESS Fls.c - add MISRA comments Specific stuff: UM - remove info regarding FlsDisableAccessToFlashControllerRegisters More information in IPV_FLASHV2 clone (ENGR00339848). Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00334148	Defect	[FLS] Incorrect CRC on flash sector configuration Problem detailed description (how to reproduce it): Please find attached a fls configuration which causes the fls driver to stop after a check of Fls_CalcCfgCRC() (see screenshot), reporting the DET error from Fls_Init: if (Fls_ConfigPtr->configCrc != Fls_CalcCfgCRC()) { #if(FLS_DEV_ERROR_DETECT == STD_ON)

Table continues on the next page...

ID	Subtype	Headline and Description
		Det_ReportError(FLS_MODULE_ID, FLS_INSTANCE_ID, FLS_INIT_ID, FLS_E_PARAM_CONFIG); Fls_ConfigPtr = NULL_PTR; #endif /* FLS_DEV_ERROR_DETECT == STD_ON */ Fls_JobResult = MEMIF_JOB_FAILED; } It seems to occur only if code-sectors are configured - data sectors work correct. Preconditions: It seems that the issue is caused by the fact that customer has not used any compiler option ?DMPC5746C or ?DMPC5748G, but this is required by FLS driver. Please remove this dependency, the derivative selection shall be done only in Resource module and not as a compiler option. 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): [...]
ENGR00334355	Defect	[FLS] MISRA violations found in generic files to be fixed Problem detailed description (how to reproduce it): There're 3 MISRA violations in the FLS not caught by ITG: - Fls_Cfg.c/Fls_PBCfg.c: The numeric values of ./FlsAcErase and ./FlsAcWrite are not stored as unsigned ints ("U") in the Fls_ConfigSetPC structure (and PB equivalents): [!WS "4"!](Fls_LLD_AcErasePtrType)[!IF "../../FlsGeneral/FlsAcLoadOnJobStart"!][!IF "normalize-space(./FlsAcErasePointer) = 'NULL_PTR' or normalize-space(./FlsAcErasePointer) = 'NULL'"!][!"normalize-space(./FlsAcErase)"!]U[!ELSE!]&[!"./FlsAcErasePointer"!][!ENDIF!][!ELSE!]&Fls_LLD_AccessCode[!ENDIF!], /* FlsAcErase */ - Fls_LLD_AddressEncode.m4 The conditional compilation must be extended to cover ENGR00328171 (Optimized ECC handling in code flash segments) as well: #if ((FLS_ECC_WITH_NO_EXCEPTION == STD_ON) && (FLS_LLD_DATA_FLASH_EXISTS == STD_ON)) 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00328058	NewWork	[FLS] Optimized ECC handling in code flash segments NewWork Description: Upon reading data from flash memory, the driver always evaluates the status of the MCR[EER] bit to check whether there were read ECC-affected data. But since reading the ECC-affected data from CODE flash segments typically leads to the IVOR exception on PPC platforms, further evaluation of EER status is not needed in such a case (unlike to the DATA flash segments). Since EER bit is set in the case when any of four 64-bit ECC segments within a 256-bit prefetch buffer page contains ECC error, skipping the EER check would allow to reach optimized 64-bit FEE virtual page size for at least CODE flash segments (instead of current 256-bit size). For devices which contains only CODE flash segments (e.g. Calypso) this would lead to the optimized 64-bit FEE virtual page for all segments. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00337517	NewWork	[FLS] Remove CER mechanism from sMCAL code base (generic file only) As per iTWG decision the CER mechanism need to be removed from sMCAL codebase
ENGR00332158	NewWork	[FLS] Review and update User manual and Integration Manual NewWork Description: Please review in all manuals the following items: - Interrupt service routine and interrupt vector are documented correctly. - The non-autosar/ Freescale specific parameters are documented with a correct description. - Remove unavailable parameters/APIs in the platform. - Deviation from ASR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00334738	NewWork	[FLS] Review and update driver FMEA for Panther NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Review and update driver FMEA for Panther
ENGR00348666	Defect	[FLS] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00333845	NewWork	[FLS] Update code based on code checklist NewWork Description: Update the code to pass the code checklist rules. Update UM, Chapter Runtime Errors (rename Fls_LLD_... to Fls_Flash_...)
ENGR00349771	Defect	[FR] Correct and optimize LPdu-based services Problem detailed description (how to reproduce it): Fr Driver is not fully compliant with the following requirements [FR224], [FR244], [FR611],[FR233] especially Bullet 1). - Functions Fr_TransmitTxLPdu, Fr_CancelTxLpdu, Fr_CheckTxLPdu do not figure out whether a buffer is mapped to the transmission (Bullet1 [FR224], [FR244], [FR611]) - Function Fr_ReceiveRxLPdu do not figure out whether a buffer is mapped to the reception (Bullet1 [FR233]) - Function Fr_CheckTxLPdu shall check CMT bit instead of MBIF. (See comment 6 FR ASR spec V2.5.0: "The returned status does not check whether a transmission has been really performed, but returns whether a transmission

Table continues on the next page...

ID	Subtype	Headline and Description
		resource is empty or not") Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional [...] Trigger: N/A Observed behavior: LPdu based services do not figure out whether buffer is mapped to the transmission or reception. Expected behavior: In case of physical buffer not mapped to the transmission for Fr_TransmitTxLPdu, Fr_CancelTxLPdu, Fr_CheckTxLPdu -> then DEM is called In case of physical buffer not mapped to the transmission for Fr_ReceiveRxLPdu ->then DEM is called. Fr_CheckTxLPdu function will check CMT bit instead of MBIF flag in the MBCCSR register. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00351065	NewWork	[FR] Create quality package for Calypso 6M RTM1.0.0 Description: Create final Test Report (if ITG is not involved in the release) Create final MISRA report (if ITG is not involved in the release) Create final compiler warnings report (if ITG is not involved in the release) Create final VSMD report Generate Traceability Matrix Report (if it is not EAR release) Create UM/IM
ENGR00344706	Defect	[FR] FR driver Fr_GetWakeupRxStatus does not reset the WUA and WUB flags Problem detailed description (how to reproduce it): In MPC564xC MCAL 4.0 RTM 1.0.1, FlexRay driver. The API Fr_GetWakeupRxStatus reads the the wakeup indication from PSR3, but it should also clear the flags (according to requirement FR588, item 2). This causes problems to the FR state management that doesn't see consistent wakeup flags in the controller and in the transceiver. Preconditions: Observed behavior: - Fr_GetWakeupRxStatus does not clear WUA, WUB flags When can it be observed? (at configuration time, at runtime, at compile time?) - At runtime Expected behavior: - WUA, WUB flags are cleared Proposed solution (Optional): Update Fr_GetWakeupRxStatus and Fr_IPW_GetWakeupRxStatus functions.
ENGR00355174	Defect	[FR] Fixing MISRA error

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): There is new misra error due to change of DET 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): Fixing misra errors
ENGR00348180	Defect	[FR] Fr_Flexray.c file linking issue in postbuild loadable implementation Problem detailed description (how to reproduce it): The post build loadable implementation, freescale drivers modules static code compiling and flashing separately from post build configuration(Fr_PBcfg). While linking Fr_flexray.c (static code) file create the linker error because of "Fr_MemoryArea" array which is present in Fr_PBcfg.c. In post build loadable implementation we are not compiling "Fr_PBcfg.c" and "Fr_Flexray.c" file together .The "Fr_MemoryArea" configuration container not present in Tresos tool (ver 14.1 090 plug-in). So please move the "Fr_MemoryArea" array declaration from "Fr_PBcfg.c" file to "Fr_flexray.c" file or suggest any other methods to avoid this linking issue. Please find attached files fixing the issue. Preconditions: Variant PostBuild - Remove Fr_PBcfg.c file from compilation Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Linker error because Fr driver uses directly &Fr_MemoryArea[0] address and the Fr_MemoryArea symbol is not known in case of Fr_Flexray.c and Fr_PBcfg.c are compiled separately. Expected behavior: Driver can be compiled separately from configuration. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Do not use &Fr_MemoryArea[0] directly in the driver. Use pointer to access FlexRay memory base address (Fr_CCHardwareConfigPtr->CCFlexRayMemoryBaseAddress) For PostBuild Variant replace: (See attached files which solves this problem) [!WS "0"!]#define FR_CC_FLEXRAY_MEMORY_BASE_ADDR ((uint32)(&Fr_MemoryArea[0])) with [!WS "0"!]#define FR_CC_FLEXRAY_MEMORY_BASE_ADDR (Fr_CCHardwareConfigPtr->CCFlexRayMemoryBaseAddress)

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00340922	NewWork	[FR] Implement support for multiple number of FlexRay CC NewWork Description: There are devices like Matterhorn, QUASAR3 with more than 1 FlexRay Communication controller. Current FR driver and FR Plugin support only one FR Controller Expected behavior: Driver and Plugin will support multiple controllers Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00336030	Defect	[FR] Missing included Dem.h in config file Problem detailed description (how to reproduce it): Due to the fact that Base plugin was updated (removed Serr.h) the Fr driver does not compiles. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: - Latest BASE plugin without Serr.h Observed behavior: - Driver is not compilable 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): add include "Dem.h" into the Fr_PBcfg.c, Fr_Cfg.c remove include "Mcal.h"
ENGR00337518	NewWork	[FR] Remove CER mechanism from sMCAL code base. As per iTWG decision the CER mechanism need to be removed from sMCAL codebase
ENGR00353042	NewWork	[FR] Review UML design against checklist NewWork Description: Review the UML design against design checklist. Implement the findings. Please attach the filled-in checklist to this CR before closing it. Checklist is available at (also attached): http://compass.freescale.net/livelink/livelink/open/215499504

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00348678	Defect	[FR] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00333342	Defect	[FR] Wrong LAST_MB_SEG1 generated in configuration Problem detailed description (how to reproduce it): - In case of no static frame is configured, then FR generation template generates out of range value -1U for the LAST_MB_SEG1 in the FR_MBSSUTR. Preconditions: - no static frame is configured Test Case ID (internal TC that caught the defect) - optional [...] Trigger: - Run generation template with no static slot configured. Observed behavior: STATIC CONST(Fr_CCBufferConfigSetType, FR_APPL_CONST) FrIfMC0_Clst0_Ctrl0_BufferCfgSet_PB = { &FrIfMC0_Clst0_Ctrl0_LPduInfoCfgSet_PB[0], /* LPdu configuration set */ &FrIfMC0_Clst0_Ctrl0_BAddress32Table_PB[0], /* Buffer Addresses Table */ &FrIfMC0_Clst0_Ctrl0_BOffset16Table_PB[0], /* Buffer Offsets Table */ 4U, /* Number of items in FrIfMC0_Clst0_Ctrl0_LPduInfoCfgSet_PB */ 1U, /* Data size in buffers segment 1 - gPayloadLengthStatic */ 127U, /* Data size in buffers segment 2 - pPayloadLengthDynMax */ -1U, /* Last MB in segment 1 (Number of MB in Segment1 - 1) */

Table continues on the next page...

ID	Subtype	Headline and Description
		3U, /* Last individual MB; (Number of MB in Segment1 + Number of MB in Segment2 - 1) */ /* Receive shadow buffers configuration */ 4U, /* Ch A, seg 1 - unused */ 4U, /* Ch B, seg 1 - unused */ 4U, /* Ch A, seg 2 - the initial buffer index */ 4U /* Ch B, seg 2 - unused */ }; 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): In case of no static frame is configured, then all MB belongs to the first segment. In such case the following value has to be used: - Data size in segment 1 =pPayloadLengthDynMax - Data size in segment 2 is 0 - LAST_MB_SEG1 = LAST_MB_UTIL Example: STATIC CONST(Fr_CCBufferConfigSetType, FR_APPL_CONST) FrIfMC0_Clst0_Ctrl0_BufferCfgSet_PC = { &FrIfMC0_Clst0_Ctrl0_LPduInfoCfgSet_PC[0], /* LPdu configuration set */ &FrIfMC0_Clst0_Ctrl0_BAddress32Table_PC[0], /* Buffer Addresses Table */ &FrIfMC0_Clst0_Ctrl0_BOffset16Table_PC[0], /* Buffer Offsets Table */ 2U, /* Number of items in FrIfMC0_Clst0_Ctrl0_LPduInfoCfgSet_PC */ 8U, /* Data size in buffers segment 1 - pPayloadLengthDynMax */ 0U, /* Data size in buffers segment 2 - second message buffer segment is empty */ 1U, /* Last MB in segment 1 */ 1U, /* Last individual MB; (All individual MBs belong to Segment1) */ /* Receive shadow buffers configuration */ 2U, /* Ch A, seg 1 - the initial index of the MB header field */ 3U, /* Ch B, seg 1 - the initial index of the MB header field */ 3U, /* Ch A, seg 2 - unused */ 3U /* Ch B, seg 2 - unused */ };
ENGR00334687	NewWork	[GPT] Change the Dem error const variable name Initial Description: If we configure Gpt_E_Forbidden_Invocation in GptDemEventParameterRefs, Gpt_Cfg.h, Gpt_Cfg.c and Gpt_PBcfg.c generate following lines. Gpt_Cfg.h: {code} extern CONST(Dem_EventIdType, GPT_CONST)Gpt_E_Forbidden_Invocation; {code} Gpt_Cfg.c: {code} CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} Gpt_PBcfg.c: {code}

Table continues on the next page...

ID	Subtype	Headline and Description
		CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} However, Gpt_E_Forbidden_Invocation will be defined like below in our Dem_IntErrId.h: {code} #define DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation 27U #define Gpt_E_Forbidden_Invocation 27U #define Dem_Gpt_E_Forbidden_Invocation 27U {code} Our EB Dem module can be used in all ASR versions and therefore is backward compatible to older ASR versions: * 1st line: for ASR 4.0.3 and up * 2nd line: for ASR 3.1 * 3rd line: for ASR 3.2 and up to ASR 4.0.2 So, because of this, the above results in like these. {code} extern CONST(Dem_EventIdType, GPT_CONST)27U; CONST(Dem_EventIdType, GPT_CONST) 27U = (Dem_EventIdType)27U; {code} And due to this, compile of Gpt_PBcfg.c fails. Problem detailed description (how to reproduce it): compile of Gpt_PBcfg.c fails. Preconditions: [...] Observed behavior: Compiler error When can it be observed? (at configuration time, at runtime, at compile time?) at compile time Expected behavior: No compiler error Reported release baseline: MPC577XM 0.9.0 Proposed solution (Optional): To prevent such name clashes a pre/post-fix could be added to this const in the code in a future release.
ENGR00348584	NewWork	[GPT] No MemMap declaration for Gpt_InitConfigPC in Gpt.c Problem detailed description (how to reproduce it): Gpt_InitConfigPC is mapped to CONFIG_DATA_UNSPECIFIED section in Gpt_Cfg.c but this constant is not mapped to CONFIG_DATA_UNSPECIFIED in Gpt.c Preconditions: Precompile configuration Observed behavior: Linker error Expected behavior: No linker error Proposed solution (Optional): Add memory mapping declaration for Gpt_InitConfigPC in Gpt.c #define GPT_START_SEC_CONFIG_DATA_UNSPECIFIED #include "MemMap.h" #if (GPT_PRECOMPILE_SUPPORT == STD_ON) /* Extern declarations of GPT Pre compile configuration from Gpt_Cfg.c */

Table continues on the next page...

ID	Subtype	Headline and Description
		/** @violates @ref GPT_C_REF_7 MISRA 2004 Rule 8.7, Global variables containing configuration */ extern CONST(Gpt_ConfigType, GPT_CONST) Gpt_InitConfigPC; #endif #define GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED #include "MemMap.h"
ENGR00318975	Defect	[GPT] Not implement inter module version checking in driver source code Problem detailed description (how to reproduce it): The module shall perform inter module checks to avoid integration of incompatible files: all included header files shall be checked by pre-processing directives. The module shall thereby verify that <MODULENAME>_AR_RELEASE_MAJOR_VERSION and <MODULENAME>_AR_RELEASE_MINOR_VERSION are identical to the expected values, where <MODULENAME> is the module abbreviation of the external module, which provides the included header file. If the values are not identical, an error shall be raised at compile time. Preconditions: [...] Test Case ID (internal TC that caught the defect) - by review [...] Trigger: [...] Observed behavior: [...] Expected behavior (example for WDG): Check Standard version: #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK #if ((WDG_CHANNEL_AR_RELEASE_MAJOR_VERSION != STD_TYPES_AR_RELEASE_MAJOR_VERSION) || \ (WDG_CHANNEL_AR_RELEASE_MINOR_VERSION != STD_TYPES_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Wdg_Channel.h and Std_Types.h are different" #endif #endif Inter check between module: #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK #if ((WDG_CHANNEL_AR_RELEASE_MAJOR_VERSION != GPT_AR_RELEASE_MAJOR_VERSION) || \ (WDG_CHANNEL_AR_RELEASE_MINOR_VERSION != GPT_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Wdg_Channel.h and Gpt.h are different" #endif #endif Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00353602	Defect	[GPT] Parameter maximum value in eMIOS's tick is redefined Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		The maximum value in eMIOS's tick in TRESOS GUI is defined 24bits. Meanwhile, the eMIOS timer width is defined 16bits in reference manual (RM) document. Proposed solution (Optional): To avoid to take mistakes when setup the maximum value in ticks. Redefine parameter maximum value ticks So, solution is the replacement: <mt:xpath expr="((. != '16777215') <mt:tst expr=">=16777215"/> with <mt:xpath expr="((. != '65535') <mt:tst expr=">=65535"/> Respectively,
ENGR00352410	Defect	[GPT] Parameters using ENABLED attribute are removed from the data model Problem detailed description (how to reproduce it): A lot of parameters (spread over all modules) that are visible within Tresos GUI, but greyed out as not editable or conditionally visible (depending on different configuration parameters) are missing within the exported AUTOSAR configuration file (see attached Tresos_GUI.jpg). After import of this configuration file the configuration validation fails, because those parameters are mandatory (multiplicity 1 .. x) within the module description file in AUTOSAR format. The export has to consider those parameters. What generates this problem is that when Tresos generates the EPD, it removes all nodes which have the attribute ENABLE=False. Proposed solution: To avoid Tresos removing AutoSAR and Vendor Specific mandatory parameters (multiplicity 1 .. x) when epd file is generated, in xdm file, their attribute ENABLE (if it exists) shall be replaced by EDITABLE attribute. The multiplicity can be observed in the .epd files in the plugin's autosar folder. So, the solution is the replacement of: <a:da name="ENABLE" type="XPath"> with <a:da name="EDITABLE" type="XPath"> This shall be done for all AutoSAR mandatory and Vendor specific parameters.
ENGR00343125	Defect	[GPT] Remove local function header definitions from Gpt.h Problem detailed description (how to reproduce it): Warnings appear due to header declarations for local functions in Gpt.h Preconditions: Use the linaro compiler Test Case ID (internal TC that caught the defect) - optional - Trigger: Compiled on Linaro compiler on Halo Observed behavior: [...] Expected behavior: [...] 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): Remove those headersfrom Gpt.h
ENGR00332161	NewWork	[GPT] Review and update User manual and Integration Manual NewWork Description: Please review in all manuals the following items: - Interrupt service routine and interrupt vector are documented correctly. - The non-autosar/ Freescale specific parameters are documented with a correct description. - Remove unavailable parameters/APIs in the platform. - Deviation from ASR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00354753	NewWork	[GPT] Update Include for Gpt.h alingned with Dem lastest NewWork Description: [...] Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00354945	Defect	[GPT] Update Misra warning Problem detailed description (how to reproduce it): [...] 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00348667	Defect	[GPT] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00349443	NewWork	[ICU] Align with MCL changes when support DMA NewWork Description: MCL driver changes DMANotification_Object name in file xdm: User callback function to report that the trasnfer is half or complete depending on configuration. Update the module to allign with these MCL changes. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00333866	NewWork	[ICU] Analyze data flow anomalies from the StaticAnalysis report NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Analyze and fix all problems for the data flow anomalies in the static analysis report. All suspicious variables usage that are not errors need to be documented in the Remarks column. More details on data flow anomalies can be found in the attachement
ENGR00346181	Defect	[ICU] Compiler warnings when using SignalMesureWithoutInterrupt Problem detailed description (how to reproduce it): Compiler warnings when call function Icu_ValidateSignalMesureWithoutInterrupt in case don't use SignalMesureWithoutInterrupt. Preconditions: NA Test Case ID (internal TC that caught the defect) - optional NA Trigger: [...] Observed behavior: Warning when build test Expected behavior: Don't have any Compiler warnings. Proposed solution (Optional): In file Icu.c: should be changed as below: function Icu_ValidateSignalMesureWithoutInterrupt should be called when (ICU_GET_PULSE_WIDTH_API == STD_ON)
ENGR00350352	Defect	[ICU] Correct DMA for Measuremode Problem detailed description (how to reproduce it): When use DMA for Measuremode, the Icu_SignalMeasurementDmaProcessing function check IcuDefaultStartEdge. It will wrong capture in Duty_Cycle and Active_Time mode. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: When use DMA, Measuremode capture righ value in Duty_Cycle and Active_Time mode. Proposed solution (Optional): In file Icu.c, Icu_SignalMeasurementDmaProcessing function should be check for Duty_Cycle and Active_Time.
ENGR00343067	Defect	[ICU] Don't check ICU_MODE_SIGNAL_MEASUREMENT when use DMA Problem detailed description (how to reproduce it): Don't check ICU_MODE_SIGNAL_MEASUREMENT when use DMA in file Icu_PluginMacros.m

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: On platforms with EMIOS, DMA is applicable for both signal measurement and timestamp measurements. In file Icu_PluginMacros.m, should be checked 2 mode when use DMA. Proposed solution (Optional): implement as below In file Icu_PluginMacros.m change from: */!]CONST(Icu_DmaReqConfigType, ICU_CONST) DmaReqConfig_[!"$VARIANT"!][!"@index"!] [[!"num:i($noOfDmaReqConfigured)"!]]=[!CR!]{[!CR!][!/* */!][!LOOP "IcuChannel/*"!][!/* */!][!IF "IcuMeasurementMode = 'ICU_MODE_TIMESTAMP'and IcuDMAChannelEnable = 'true'"!][!/* to: */!]CONST(Icu_DmaReqConfigType, ICU_CONST) DmaReqConfig_[!"$VARIANT"!][!"@index"!] [[!"num:i($noOfDmaReqConfigured)"!]]=[!CR!]{[!CR!][!/* */!][!LOOP "IcuChannel/*"!][!/* */!][!IF "IcuDMAChannelEnable = 'true'"!][!/*
ENGR00354862	Defect	[ICU] Fix mirsa violations regards Det_ErrorReport Problem detailed description (how to reproduce it): Mirsa violations Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: Don't have any Misra violations Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Fix mirsa violations regards Det_ErrorReport
ENGR00346143	Defect	[ICU] IcuEmiosBusSelect container in eMios platforms gives errors when generating Problem detailed description (how to reproduce it): IcuEmiosBusSelect Container in eMios platforms gives errors when generating Preconditions: Platforms have

Table continues on the next page...

ID	Subtype	Headline and Description
		Test Case ID (internal TC that caught the defect) - optional NA Trigger: Creating the pluggin with Tresos Pluggin Builder and/or generating .epds on platforms with eMios (tested on Calypso and Cobra55) Observed behavior: Error when generating the pluggin or generating tresos epds. Expected behavior: There shouldn't be errors. Proposed solution (Optional): The ENABLE field should be replaced by a default value like EMIOS_BUS_A or EMIOS_BUS_INTERNAL_COUNTER, even if it's not the recomended default buss source, due to the eMios Architecture. This should also lead to an extra paragraph in the manual regarding the default buss source.
ENGR00334428	Defect	[ICU] Icu_DeInit doesn't clear the SIUL2_IFCPR register Problem detailed description (how to reproduce it): The Icu_DeInit function doesn't clear the SIUL2 IFCPR Register (Input Filter Register) Preconditions: Icu_DeInit is called Test Case ID (internal TC that caught the defect) - optional - none Trigger: [...] Observed behavior: IFCPR register is written with the init value Expected behavior: IFCPR register 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): Update the code in Icu_IPW.c to call the SIUL2 function with the parameter 0 if the globalInit parameter is FALSE
ENGR00348772	Defect	[ICU] Icu_GetInputState and Icu_GetDutyCycles relationship Initial Description: The customer reported that the function Icu_GetDutyCycles call Icu_ClearBitChState so the Icu_GetInputState cannot get the correct input state after Icu_GetDutyCycles is called. Problem detailed description (how to reproduce it): Preconditions: [...] Observed behavior: Icu_GetInputState does not return ICU_ACTIVE if Icu_GetDutyCycles is called in perior. When can it be observed? (at configuration time, at runtime, at compile time?) Runtime Expected behavior: According to Icu_GetInputState spec: Icu_InputStateType

Table continues on the next page...

ID	Subtype	Headline and Description
		ICU_ACTIVE: An activation edge has been detected ICU_IDLE: No activation edge has been detected since the last call of Icu_GetInputState() or Icu_Init(). Reported release baseline: MPC577XM MCAL1.0.0 Proposed solution (Optional): Just check Icu_GetInputState and IcuGetDutyCycles spec, there is no relationship explicitly stated in the specification but it seems these functions have some relations. e.g: [ICU314] For Signal Measurement a channel should be set to ICU_ACTIVE not until this measurement has completed and the driver is able to provide useful information on the input signal. And [ICU344] The function Icu_GetDutyCycleValues shall return "0" in case captured duty cycle values were already returned once by this service and this service is called again (similar to Figure 9.19, letter "D") However, it is still uncleared.
ENGR00321976	NewWork	[ICU] Implement delta PR-MCAL-3297 requirements Align to CPRD 1.23 the following requirements need to be implemented 1) PR-MCAL-3297 "The ICU driver shall provide an optional configuration option to use DMA for ICU channel read by using DMA minor loop for data transfer and DMA major loop for buffer full signaling. By default this optional functionality shall be disabled. Restrictions: a) The DMA support was added only, to reduce the number of interrupts but not to add additional functionalities such as saving more then one samples for duty cycles and period. b) On platforms with eTimer, this requirement is applicable for timestamp measurements only, because the signal measurement is done on these platforms using only one interrupt (the hardware provides two capture registers that can be handled in a way that the number of interrupts for a measurement is '1'). c) On platforms with EMIOS, this requirement is applicable for both signal measurement and timestamp measurements."
ENGR00331973	NewWork	[ICU] Implement requirement PR-MCAL-3275.icu on platforms with eMIOS NewWork Description: Implement requirement PR-MCAL-3275.icu on platforms with eMIOS Expected behavior: This should be done as an extra checkbox in the pluggin that will add the SM posibility without interrupt Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): NA
ENGR00328871	NewWork	[ICU] Implement requirement PR-MCAL-3320

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: Add function Icu_GetCaptureRegisterValue Expected behavior: [...] Requirement source: PR-MCAL-3320 "The Icu driver shall provide an optional function, API and configuration parameters for reading the capture register content of a ICU channel. Per default, this functionality shall be disabled." (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00328660	NewWork	[ICU] Implement requirement PR-MCAL-3327 NewWork Description: Implement requirement PR-MCAL-3327 "The ICU driver shall execute the following service request only, if it is NORMAL mode: - Icu_EnableWakeup - Icu_StartTimestamp - Icu_EnableEdgeCount - Icu_EnableEdgeDetection - Icu_StartSignalMeasurement Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00349846	NewWork	[ICU] No MemMap declaration for Icu_Configs_PC in Icu.c Problem detailed description (how to reproduce it): Icu_Configs_PC is mapped to CONFIG_DATA_UNSPECIFIED section in Icu_Cfg.c but this constant is not mapped to CONFIG_DATA_UNSPECIFIED in Icu.c Preconditions: Precompile configuration Observed behavior: Linker error Expected behavior: No linker error Proposed solution (Optional): Add memory mapping declaration for Gpt_InitConfigPC in Gpt.c #define ICU_START_SEC_CONFIG_DATA_UNSPECIFIED #include "MemMap.h" #ifdef ICU_PRECOMPILE_SUPPORT /* @violates @ref Icu_c_REF_8 MISRA 2004 Required Rule 8.7, objects shall be

Table continues on the next page...

ID	Subtype	Headline and Description
		* defined at block scope */ extern CONST(Icu_ConfigType, ICU_CONST) Icu_Configs_PC; #endif #define ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED #include "MemMap.h"
ENGR00318976	Defect	[ICU] Not implement inter module version checking in driver source code Problem detailed description (how to reproduce it): The module shall perform inter module checks to avoid integration of incompatible files: all included header files shall be checked by pre-processing directives. The module shall thereby verify that <MODULENAME>_AR_RELEASE_MAJOR_VERSION and <MODULENAME>_AR_RELEASE_MINOR_VERSION are identical to the expected values, where <MODULENAME> is the module abbreviation of the external module, which provides the included header file. If the values are not identical, an error shall be raised at compile time. Preconditions: [...] Test Case ID (internal TC that caught the defect) - by review [...] Trigger: [...] Observed behavior: [...] Expected behavior (example for WDG): Check Standard version: #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK #if ((WDG_CHANNEL_AR_RELEASE_MAJOR_VERSION != STD_TYPES_AR_RELEASE_MAJOR_VERSION) || \ (WDG_CHANNEL_AR_RELEASE_MINOR_VERSION != STD_TYPES_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Wdg_Channel.h and Std_Types.h are different" #endif #endif Inter check between module: #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK #if ((WDG_CHANNEL_AR_RELEASE_MAJOR_VERSION != GPT_AR_RELEASE_MAJOR_VERSION) || \ (WDG_CHANNEL_AR_RELEASE_MINOR_VERSION != GPT_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Wdg_Channel.h and Gpt.h are different" #endif #endif Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00334813	NewWork	[ICU] Remove the Dem error const variable name Cloned from Gpt. Same logic, as below, applies to ICU also. Initial Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		If we configure Gpt_E_Forbidden_Invocation in GptDemEventParameterRefs, Gpt_Cfg.h, Gpt_Cfg.c and Gpt_PBcfg.c generate following lines. Gpt_Cfg.h: {code} extern CONST(Dem_EventIdType, GPT_CONST)Gpt_E_Forbidden_Invocation; {code} Gpt_Cfg.c: {code} CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} Gpt_PBcfg.c: {code} CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} However, Gpt_E_Forbidden_Invocation will be defined like below in our Dem_IntErrId.h: {code} #define DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation 27U #define Gpt_E_Forbidden_Invocation 27U #define Dem_Gpt_E_Forbidden_Invocation 27U {code} Our EB Dem module can be used in all ASR versions and therefore is backward compatible to older ASR versions: * 1st line: for ASR 4.0.3 and up * 2nd line: for ASR 3.1 * 3rd line: for ASR 3.2 and up to ASR 4.0.2 So, because of this, the above results in like these. {code} extern CONST(Dem_EventIdType, GPT_CONST)27U; CONST(Dem_EventIdType, GPT_CONST) 27U = (Dem_EventIdType)27U; {code} And due to this, compile of Gpt_PBcfg.c fails. Problem detailed description (how to reproduce it): compile of Gpt_PBcfg.c fails. Preconditions: [...] Observed behavior: Compiler error When can it be observed? (at configuration time, at runtime, at compile time?) at compile time Expected behavior: No compiler error Reported release baseline: MPC577XM 0.9.0 Proposed solution (Optional): To prevent such name clashes a pre/post-fix could be added to this const in the code in a future release.
ENGR00332162	NewWork	[ICU] Review and update User manual and Integration Manual NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Please review in all manuals the following items: - Interrupt service routine and interrupt vector are documented correctly. - The non-autosar/ Freescale specific parameters are documented with a correct description. - Remove unavailable parameters/APIs in the platform. - Deviation from ASR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00337175	Defect	[ICU] The duty cycle and period are not calculated correctly in case counter overflow for Signal measurement channel Problem detailed description (how to reproduce it): The duty and cycle are not correctly measured in overflow case Expected behavior: The duty and cycle are correctly measured. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In Icu_Cfg.h Replace line #define EMIOS_COUNTER_MASK ((Icu_ValueType)0xFFFFFFU) follow below: #define ICU_EMISO_24BIT_CAPABILITY (STD_ON) #if ICU_EMISO_24BIT_CAPABILITY == STD_ON #define EMIOS_COUNTER_MASK ((Icu_ValueType)0xFFFFFFU) #else #define EMIOS_COUNTER_MASK ((Icu_ValueType)0xFFFFU) #endif 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): [...]
ENGR00347631	Defect	[ICU] Update Mirsa violations Problem detailed description (how to reproduce it): MISRA errors

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: No Mirsa Violations Proposed solution (Optional): Correct MISRA violations
ENGR00331621	Defect	[ICU] Update comment misra and compiler warning Problem detailed description (how to reproduce it): Misra and compiler warning when build test Preconditions: Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: Update code and comment follow Misra standard Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00349457	Defect	[ICU] Update define regarding Icu_GetPulseWidth function Problem detailed description (how to reproduce it): When compile driver, error appear. This Icu_GetPulseWidth function must be called when ICU_SIGNAL_MEASUREMENT_API= STD_ON. This change should be ported for platforms with eMIOS. 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): The Icu_GetPulseWidth function should be compile when both ICU_GET_PULSE_WIDTH_API and ICU_SIGNAL_MEASUREMENT_API are STD_ON

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00333602	NewWork	[ICU] Update defines for changed requirement PR-MCAL-3297.icu NewWork Description: Change the previous ICU_MEASUREMENTS_USE_DMA define (that only applied for Timestamp mode) to ICU_TIMESTAMP_USES_DMA and ICU_SIGNALMEASUREMENT_USES_DMA. - the last one will only be active for platforms with eMios. Expected behavior: The DMA to work indepedently for both modes - on platforms with eMios and to work for timestamp - on other platforms Requirement source: None - internal rework Proposed solution (Optional): See NewWork Description
ENGR00347700	Defect	[ICU] WKPU single interrupt define is invalid Initial Description: Issue in driver configuration files regarding the WKPU IP (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): ICU_WKPU_SINGLE_INTERRUPT is defined in Icu_Cfg.h, causing a wrong ISR approach in Icu_Wkpu_Irq.c. In the Calypso reference manual, there are 4 interrupt lines. This issue causes the remaining 3 WKPU ISR's to not be used (8-15, 16-23, 24-31) Preconditions: Driver uses the WKPU IP 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...)

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00331614	NewWork	[ICU]Remove unneeded traceability marker in Icu_SIUL2_Irq.c NewWork Description: Update code in file Icu_SIUL2_irq.c for run Tracer Matric Expected behavior: In Icu_SIUL2_irq.c - At line 318: remove * @implements Icu_Siul2_ProcessSingleInterrupt Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00328997	Defect	[ADC] Fix Misra errors for IPV_ADCDIG files Problem detailed description (how to reproduce it): Fix Misra errors for IPV_ADCDIG files 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): [...]
ENGR00329701	Defect	[ADC] Fix implementation of requirement CPR-MCAL-720 - HW triggered groups should not be removed from the queue Problem detailed description (how to reproduce it): Fix implementation of requirement CPR-MCAL-720 - HW triggered groups should not be removed from the queue. Hw triggered groups are not stopped implicitly. 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00347925	Defect	[ADC] The SW normal queue is checked twice at the end of Adc_Adcdig_DmaEndNormalConv Problem detailed description (how to reproduce it): At the end of Adc_Adcdig_DmaEndNormalConv, the SW normal queue is checked twice, the next SW group is started twice. 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): [...]
ENGR00328716	Defect	[ADC] The function Adc_Adcdig_HwResultReadGroup alway returns E_NOT_OK if it is invoked with a Group that contains duplicated channels Problem detailed description (how to reproduce it): The Group 0 is configured as hw trigger. In the container AdcGroupDefinition: AN0, AN1, AN1, AN2. [...] Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional Adc_TC_0130 [...] 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): [...]
ENGR00331087	NewWork	[ADC] Add code to support hardware injected trigger by BCTU NewWork Description: Add code to support hardware injected trigger by BCTU in the CTU trigger mode 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): [...]
ENGR00331044	Defect	[ADC] BCTU control mode for 3 units does not work Problem detailed description (how to reproduce it): BCTU control mode for 3 units does not work. The results in the trigger buffers are not as expected. 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): [...]
ENGR00352737	Defect	[ADC] Bctu notification lost Problem detailed description (how to reproduce it): The problem is bctu notification is lost by this line: if (u8TotalIndex > (pTriggerPtr->u8NumListElement)) u8TotalIndex is updated to equal number of channels converted. So the notification must be called as soon as it equal to u8NumListElement. 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): [...]
ENGR00327958	Defect	[ADC] If DMA transfer and BCTU triggering are used the destination address must be re-configured after each chain conversion completed Problem detailed description (how to reproduce it): In the DMA mode, the destination address must be re-configured after each

Table continues on the next page...

ID	Subtype	Headline and Description
		chain conversion completed 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): [...]
ENGR00329111	Defect	[ADC] List channels will be overwritten if Adc_EnableCtuTrigger is called more than once for different unit Problem detailed description (how to reproduce it): List channels will be overwritten if Adc_EnableCtuTrigger is called more than once for different unit 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): [...]
ENGR00347376	NewWork	[MCU] Add support for CMU_ISRbit NewWork Description: Add support for detecting frequency less than reference clock event status. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00341879	Defect	[MCU] Mcu_CMU_DeInitdoes not clear CMU_MDR to reset value. Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		In Mcu_CMU_DeInit, CMU_MDR is cleared by macro REG_BIT_CLEAR32(MC_CMU_MDR_X_ADDR32(u8IndexCmu), MCU_CMU_MDR_MD_RESET_VALUE_U32); where #define MCU_CMU_MDR_MD_RESET_VALUE_U32 ((uint32)0x00000000U) If CMU_MDR contains any data different from 0, It will not be cleared to the reset 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): [...]
ENGR00352051	NewWork	[MCU] add support for Autosar standard 4.2.1 NewWork Description: Add support in MCAL for Autosar 4.2.1 standard. First target is a code drop for Cobra55. We need to be able to generate releases for both ASR4.0.3 and ASR4.2.1 standards using the files in SASW vob. Expected behavior: Implement deltas to support the ASR 4.2.1 standard Requirement source: Customer request (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Review and implement changes in code to support Autosar 4.2.1 standard.
ENGR00331639	NewWork	[SPI] Add end job notification in synctransmit mode Problem detailed description (how to reproduce it): Add the code to identify the current job was end: /* Perform job EndNotification (if there is one) */ if (NULL_PTR != pJobConfig->EndNotification) { pJobConfig->EndNotification(); } else { /* Do nothing */ } Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] 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): [...]
ENGR00346899	NewWork	[SPI] Add no_cacheable definition to header file base on Diab compiler behavior NewWork Description: When we run test with Diab compiler and D-Cache was enable, the transmit and receive buffers which were used in DMA mode, was allocated in cache memory area. Expected behavior: All buffers were allocated to no_cache memory area Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00334080	NewWork	[SPI] Code review against checklist NewWork Description: Review code against code checklist template and correct findings. The initial and final code checklist filled-in files shall be attached to the CR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00341009	Defect	[SPI] Configured default transmit value is not sent as expected Problem detailed description (how to reproduce it): Spi_SetupEB is used with a NULL pointer to cover the following Autosar requirement : SPI028: When the SPI Handler/Driver's environment is calling the function Spi_SetupEB with the parameter SrcDataBufferPtr being a Null pointer, the function shall transmit the default transmit value configured for the channel after a Transmit method is requested. (See also [SPI035]) . The use case is the following: Send an asynchronous Sequence with one job and two channels with EB : the first one is a 1xbyte, and the second one is a 3xWord.

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: A sequence that contains an asynchronous job, that contains 2 EB channels, as configured below. - Channel 5 : 1 byte, default value is 0x05 - Channel 11 : 2 bytes, default value is 0x0001 Observed behavior: The first sent byte is correct (0x05), but three next word are sent with invalid value (0x0005 instead of 0x0001) Note that If the channels are used in the inverted order (first 11, then 5), the issue does not appear. When can it be observed? (at configuration time, at runtime, at compile time?) at runtime. Expected behavior: Send the configured default transmit values correctly independent of the channels order/position. Proposed solution (Optional): [...]
ENGR00347816	Defect	[SPI] Fix MISRA error Problem detailed description (how to reproduce it): We have some MISRA comments wrong in Spi_DSPI.h 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): [...]
ENGR00331292	Defect	[SPI] Fix misra error Problem detailed description (how to reproduce it): The last report about Misra contains some errors. These errors must be corrected. 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.)

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution (Optional): [...]
ENGR00330517	NewWork	[SPI] Support for ITSB in SPI module The customer needs the ITSB mode in order to implement a micro second channel communication.
ENGR00347178	Defect	[ICU] Dual clock mode features works incorrectly Problem detailed description (how to reproduce it): when write Prescaler bits (UCPRE) to register to divide clock but it doesn't work as expected Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Test failed Expected behavior: Dual clock mode features works correctly. Proposed solution (Optional): In Icu_eMios_SetPrescaler function: should be clear UCPRE field before write Prescaler value.
ENGR00346693	Defect	[ICU] Fix Misra violations Problem detailed description (how to reproduce it): Misra violations Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: Generate misra report. Proposed solution (Optional): Fix Misra violations
ENGR00354571	Defect	[ICU] Status flags aren't cleared in case of a spurious interrupt Problem detailed description (how to reproduce it): The Status flags aren't cleared after spurious interrupt occured. Preconditions: Driver uses IPV_eMios Test Case ID (internal TC that caught the defect) - optional [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		Trigger: [...] Observed behavior: Test case failed. Expected behavior: The status flags must be cleared in this case. Proposed solution (Optional): Change file eMios_Commom.c: Interrupt handler for eMios channels: first check match event occured then check that a match event generates an interrupt and interrupts are enabled. if interrupts aren't enabled, must cleard status flags.
ENGR00336860	Defect	[ICU] The driver calculates wrong value for the multiple channels configured in Signal Measurement Mode. Problem detailed description (how to reproduce it): The static variables in function Icu_eMios_SignalMeasurement are used for multiple channels. static VAR(Icu_ValueType, AUTOMATIC) CapturedActivePulseWidth = (Icu_ValueType) 0U; static VAR(Icu_ValueType, AUTOMATIC) TimeStart = (Icu_ValueType) 0U; These will lead to wrong result in some case if the variables are updated by other channels interrupt. Preconditions: Some signal measurement are started concurrent. Observed behavior: Wrong period measured. Expected behavior: The period and duty cycle are measured correctly Proposed solution (Optional): In Icu_eMios.c add two global variable (array): /** *@brief Array for saving the timer start value */ volatile VAR(Icu_ValueType, ICU_VAR) Icu_TimeStart[ICU_MAX_CHANNEL]; /** *@brief Array for saving the previous pulse with */ volatile VAR(Icu_ValueType, ICU_VAR) Icu_CapturedActivePulseWidth[ICU_MAX_CHANNEL]; In Icu_eMios_SignalMeasurement: Remove 2 declare static VAR(Icu_ValueType, AUTOMATIC) CapturedActivePulseWidth = (Icu_ValueType) 0U; static VAR(Icu_ValueType, AUTOMATIC) TimeStart = (Icu_ValueType) 0U; Instead of variable CapturedActivePulseWidth by value Icu_CapturedActivePulseWidth[hwChannel], Instead of variable TimeStart by value Icu_TimeStart[hwChannel] Init two variable global in Icu_eMios_StartSignalMeasurement: Icu_TimeStart[hwChannel] = 0U; Icu_CapturedActivePulseWidth[hwChannel] = 0U; Plase see attach file for detail
ENGR00336852	Defect	[ICU] The duty cycle and period are not calculated correctly in case counter

Table continues on the next page...

ID	Subtype	Headline and Description
		overflow for Signal measurement channel Problem detailed description (how to reproduce it): According to MPC5748G Rev3, The EMIOS of MPC5748G has 16-bit counter however it is defined as 24-bit mask in Icu_Cfg.h. Observed behavior: The duty and cycle are not correctly measured. Expected behavior: The duty and cycle are correctly measured. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): In Icu_Cfg.h Replace #define EMIOS_COUNTER_MASK ((Icu_ValueType)0xFFFFFFU) by #define EMIOS_COUNTER_MASK ((Icu_ValueType)0xFFFFU) In Icu_eMios.c For IPWM duty cycle measurement. Replace Line 886-887 IcuPeriod = (Icu_ValueType)\ (tempB + ((uint32)Bus_Period - TimeStart)); by IcuPeriod = (Icu_ValueType)\ (tempB + ((uint32)Bus_Period - TimeStart))+1; Replace Line 920 CapturedActivePulseWidth = (Icu_ValueType)(tempA + ((uint32)Bus_Period - tempB)); by CapturedActivePulseWidth = (Icu_ValueType)(tempA + ((uint32)Bus_Period - tempB)) + 1; For the other mode measurement, plus 1 to Period or Low time, High time in case of overflow. Please see the attached Icu_eMios_new.c and Icu_eMios_old.c for more details. Port.xdm and Icu.xdm are customer configuration.
ENGR00336902	Defect	[ICU] The timestamp value could be wrong for multiple channels with different interrupt priorities Problem detailed description (how to reproduce it): The static variables Icu_eMios_aBufferPtr is shared between multiple channels. If the interrupt priorities are configured differently among channels the value of low priority channel could be overwritten by the high priority one. STATIC VAR(Icu_ValueType,AUTOMATIC) Icu_eMios_aBufferPtr[4]; VAR(uint8, AUTOMATIC) u8capturedWords = 1U; VAR(uint8, AUTOMATIC) u8Index = 0U; for (u8Index = 0U; u8Index < u8capturedWords; u8Index++) { /* Store timestamp */ /** @violates @ref Icu_eMios_c_3, Cast from unsigned long to pointer.*/ Icu_eMios_aBufferPtr[u8Index]= (Icu_ValueType)Icu_eMios_GetCaptureRegA(hwChannel); } Icu_IPW_Timestamp (

Table continues on the next page...

ID	Subtype	Headline and Description
		eMIOS_ICU_CHANNEL(hwChannel), u8capturedWords,Icu_eMios_aBufferPtr, bOverflow); Proposed solution (Optional): Move to global variable for each channel.
ENGR00331061	NewWork	[ICU] Update feature DMA for SignalMeasurement Mode NewWork Description: Update feature DMA for SignalMeasurement Mode Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00346721	Defect	[PWM] EMIOS2 is not supported Problem detailed description (how to reproduce it): The customer configured the EMIOS2 Channels 10 - 20 using EB-Tresos and it is generating correctly. But while compiling the software it is throwing some errors saying PWM_EMIOS_M2_C10 to PWM_EMIOS_M2_C20 is not declared. Checked the Pwm_eMios_Types.h and found that, the channel macros for EMIOS module 2 is not defined in there. Observed behavior: Compiler Error Expected behavior: No compiler error, the configured channels of EMIOS2 work as expected. Proposed solution (Optional):
ENGR00351988	Defect	[PWM] PWM not possible to set a PWM output with 1 tick duty Initial Description: Hello, Below is an inquiry from a customer regarding PWM driver. Product: MCP574xG MCAL 0.9.0 + HF4 The customer has configured a PWM period of 100 ticks (example) and wants to set a pulse with the minimum width (1 clock tick), offset is configured at zero (the channel is configured in OPWMB). But when calling Pwm_SetDutyCycle (with 328 as DutyCycle), both the eMios registers A and B are set to 1, so there is no pulse. In fact inside function Pwm_eMios_SetRegs_OPWMB_mode, there is a +1 offset : Pwm_eMios_UpdateChannelAB(u8mIdx, u8chIdx, (uint32)(nOffset + (uint32)1U), (uint32)u32TempDutyOffset); Why do we have this +1, why is it not possible to have 0 in register A, in order to get the minimum pulse width? CE's comment: It seems +1 is to align with the internal counter as stated in UM: 3.5 eMIOS specific implementation details

Table continues on the next page...

ID	Subtype	Headline and Description
		We have removed +1 from register A, in OPWMB mode This issue also affected to OPWMT mode
ENGR00350577	Defect	[PWM] Pwm_eMios_Init do not need call DeInit Problem detailed description (how to reproduce it): Follow the ticket ENGR00280455. We should de-init all the registry before init, because the registry maybe used by external application of user (bootloader,...). However in EMIOS, the registry can be init without the need of de-init Proposed solution (Optional): we do not need call to the Pwm_eMios_Local_DeInit in Pwm_eMios_Init The Pwm_eMios_Local_DeInit can be remove, and the content of it can be put to Pwm_eMios_DeInit
ENGR00330516	Defect	[PWM] Re-calculate trigger delay when SetCounterBus is called with new period Problem detailed description (how to reproduce it): When SetCounterBus use with new period, the trigger delay have to re-calculate if OffsetAutoAdjust config is TRUE, however the trigger delay is not re-calculated Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):
ENGR00331864	Defect	[PWM] There are compilers warnings in Pwm_eMios.c Problem detailed description (how to reproduce it): In the current Pwm_eMios.c file, there are some warnings when build with DIAB compiler: 717 (etoa:2273): suspicious truncation of a 32-bit value when assigned to a 64-bit integral type (potential portability problem) 718 (etoa:5387): explicit conversion of a 64-bit integral type to a smaller integral type (potential portability problem) 1659 (etoa:5387): explicit conversion of a 64-bit integral type to a smaller integral type (potential portability problem) 1683 (etoa:5387): explicit conversion of a 64-bit integral type to a smaller integral type (potential portability problem) 2589 (etoa:2273): suspicious truncation of a 32-bit value when assigned to a 64-bit integral type (potential portability problem) 2590 (etoa:2273): suspicious truncation of a 32-bit value when assigned to a 64-bit integral type (potential portability problem) 2593 (etoa:2273): suspicious truncation of a 32-bit value when assigned to a 64-bit integral type (potential portability problem) 2594 (etoa:2273): suspicious truncation of a 32-bit value when assigned to a 64-bit integral type (potential portability problem) Preconditions: [...] 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): [...]
ENGR00344961	Defect	[PWM] Typedef of Dead time is not correct Problem detailed description (how to reproduce it): In Rainier and Cobra, the counter is 24 bits, but the Dead Time is defined is 16 bits, so it is not correct Proposed solution (Optional): The dead time should be defined following the PeriodType
ENGR00354483	Defect	[PWM]Setup DAOC mode should be put under critical section Problem detailed description (how to reproduce it): In DAOC mode, the interrupt function access to A and B register, and some other functions (SetPeriodAndDuty and SetDutyCycle) also access to these registers So we need add critical section on these function (SetPeriodAndDuty and SetDutyCycle Proposed solution (Optional): in Pwm_eMios_SetDutyCycle function SchM_Enter_Pwm_PWM_EXCLUSIVE_AREA_00(void); Pwm_eMios_SetRegs_DAOC_mode(nPeriod, u16DutyCycle, peMiosChannelConfig); SchM_Exit_Pwm_PWM_EXCLUSIVE_AREA_00(void); In Pwm_eMios_SetPeriodAndDuty SchM_Enter_Pwm_PWM_EXCLUSIVE_AREA_01(void); Pwm_eMios_SetRegs_DAOC_mode(nPeriod, u16DutyCycle, peMiosChannelConfig); SchM_Exit_Pwm_PWM_EXCLUSIVE_AREA_01(void);
ENGR00343717	NewWork	[FLS] Add IP version definition NewWork Description: Add IPV_FLASHV2 define for Cobra55. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00350325	NewWork	[FLS] Add support for Fault Injection Test point NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Add support in MCAL drivers for Fault Injection Test points. Details in attached presentation and on page: http://zcz01web02:1080/wiki/vav/Wiki%20Pages/ TestDesignWithFaultInjection.aspx Expected behavior: Updated drivers to add fault injection points Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): -Add in Base the following defines: PUT_IN_QUOTES, MCAL_FAULT_INJECTION_POINT -> they should be empty in case fault injection tests are not compiled (using define ?d ENABLE_FAULT_INJECTION) -In test files (tickets for ITG) In the fault injection tests make files we need to have ?d ENABLE_FAULT_INJECTION -In driver code (tickets for developers) Add label in driver code (tester and developer should agree on the label name and the place in the code) MCAL_FAULT_INJECTION_POINT (LABEL_NAME); Note: LABEL_NAME should be compliant to the following naming rule: <MDL>_FIP_<COUNT>_<COMMENT> Eg: FLS_FIP_1_UPDATE_VARIABLE_A
ENGR00334777	Defect	[FLS] Comment compiler warnings Problem detailed description (how to reproduce it): The following warnings were detected in DIAB (etoa:4550): variable "Fls_LLD_isDsiException" was set but never used STATIC VAR(boolean, FLS_VAR) Fls_LLD_isDsiException = (boolean)FALSE; (etoa:4177): function "Fls_LLD_EncodeReadAddr" was declared but never referenced STATIC FUNC(uint32, FLS_CODE) Fls_LLD_EncodeReadAddr In some configurations, these elements are not used and excluded from the code by conditional compilation (FLS_ECC_WITH_NO_EXCEPTION != STD_ON). Please comment them appropriately. 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): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00355375	NewWork	[FLS] Fix misra error 16.10 for Det_ReportError NewWork Description: Returned value for Det_ReportError function is not read. As a result, MISRA error is raised for Rule 16.10 ("If a function returns error information, then that error information shall be tested") Proposed solution: Comment MISRA error. Det_ReportError function always return E_OK error code, so there is no reason to test it.
ENGR00338015	Defect	[FLS] Resolve compiler warning Problem detailed description (how to reproduce it): Resolve compiler warning: 1605 (etoa:4177): function "Fls_Flash_InvalidateReadBuffers" was declared but never referenced 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): [...]
ENGR00339252	NewWork	[MCU] Add support for version 0x84803202UL NewWork Description: Add support for version 0x84803202UL Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00347301	Defect	[CAN] CLKSRC is not available on MPC5746R Problem detailed description (how to reproduce it): CLKSRC bit is not available on MPC5746R. However it is set in Can_FlexCan_ChangeBaudrate. The bit CLKSRC is always generated by CAN plugin since the clock is always selected from BUS.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00347171	Defect	[CAN] Can driver wrongly set interrupt mode Enable in Can_FlexCan_InitVariables Problem detailed description (how to reproduce it): In Can_Flexcan.c, line 578-579: /* Set interrupt mode status to interrupt disabled mode. */ Can_ControllerStatuses[u8CtrlIndex].u8InterruptMode = (uint8)CAN_INTERRUPT_ENABLED; According to the comment, the Interrupt Mode should be CAN_INTERRUPT_DISABLED but it is incorrectly set to CAN_INTERRUPT_ENABLED 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): [...]
ENGR00336623	Defect	[CAN] Can_FlexCan_Init clear the reserved region. Problem detailed description (how to reproduce it): According to 47.4.24 Error Injection Address Register (CAN_ERRIAR), there are some reserved region in FLEXCAN RAM memory. However the Can_FlexCan_Init function clears also the reserved area. i=FLEXCAN_MEMORY_START_U32; while(i<=FLEXCAN_MEMORY_END_U32) { /* @violates @ref Can_Flexcan_c_REF_4 Violates MISRA 2004 Required Rule 11.1, cast from unsigned long to pointer. */ REG_WRITE32((CAN_GET_BASE_ADDRESS(u8HwOffset)+i), 0x00U); i=i+4U; } Where FLEXCAN_MEMORY_START_U32 is defined as 0x80 and FLEXCAN_MEMORY_END_U32 as 0xADF. 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00324904	Defect	[CAN] Compiler warning/error: PduInfoType *" is incompatible with parameter of type "const Can_PduType Problem detailed description (how to reproduce it): The parameters of execution of function CanIf_CancelTxConfirmation(u32PduId , &CbkPduInfo) in Can driver is different to the specified SWS requirement [CANIF101]: void CanIf_CancelTxConfirmation(PduIdType CanTxPduId, const PduInfoType* PduInfoPtr) That means that PduInfoPtr has to be of type P2CONST(PduInfoType, AUTOMATIC) but in the current code it is used of type VAR(PduInfoType, AUTOMATIC). Preconditions: MPC56XXL_MCAL4_0_RTM_HF3_1_0_1 Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00331489	NewWork	[CAN]Create new ISR function to manage FIFO and MB events together NewWork Description: Because on Rainier cut2 the MB groups are changed it is needed to create a new ISR function to manage the FIFO and also the MB interrupt when both are used. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00347175	Defect	[CAN] Exclusive Area may not be exited in certain timing Problem detailed description (how to reproduce it): In the function Can_FlexCan_EnableControllerInterrupts, the s8IntDisableLevel is tested before entering the critical section. If the current s8IntDisableLevel =1, then the test is done then it passed, before entering the critical section, the task is preempted by higher priority task, then the Can_FlexCan_EnableControllerInterrupts is called again. s8IntDisableLevel is still 1, then the core enters the critical section. After executing s8IntDisableLevel--, s8IntDisableLevel =0. Then it exits critical section by line 2356.

Table continues on the next page...

ID	Subtype	Headline and Description
		After the higher priority task finishes, the core enters the critical section again, after executing s8IntDisableLevel--, s8IntDisableLevel =-1. The test at line 2354 is not true so the test at line 2441 is checked. This test is not true since s8IntDisableLevel is now -1. This means the exclusive area does not exit. if (Can_ControllerStatuses[Controller].s8IntDisableLevel > (sint8)0) { /* When s8IntDisableLevel is greater than 0 means Can_DisableControllerInterrupts was called at least once before this call of current function. */ SchM_Enter_Can_CAN_EXCLUSIVE_AREA_01(); /* Decrement the nesting level. */ Can_ControllerStatuses[Controller].s8IntDisableLevel--; u8DisCalledBefore = (uint8)0x1U; } /* Can_DisableControllerInterrupts and Can_EnableControllerInterrupts do not modify the software flags. */ /* (CAN208) The function Can_EnableControllerInterrupts shall perform no action when Can_DisableControllerInterrupts has not been called before. */ if (((sint8)0 == Can_ControllerStatuses[Controller].s8IntDisableLevel) && ((uint8)0x1U == u8DisCalledBefore)) { SchM_Exit_Can_CAN_EXCLUSIVE_AREA_01(); /* Processing.....*/ } else { if (Can_ControllerStatuses[Controller].s8IntDisableLevel > (sint8)0) { SchM_Exit_Can_CAN_EXCLUSIVE_AREA_01(); } } Preconditions: Proposed solution (Optional): Enter the critical section before the test of s8IntDisableLevel SchM_Enter_Can_CAN_EXCLUSIVE_AREA_01(); if (Can_ControllerStatuses[Controller].s8IntDisableLevel > (sint8)0) { /* When s8IntDisableLevel is greater than 0 means Can_DisableControllerInterrupts was called at least once before this call of current function. */ /* Decrement the nesting level. */ Can_ControllerStatuses[Controller].s8IntDisableLevel--; u8DisCalledBefore = (uint8)0x1U; } /* Can_DisableControllerInterrupts and Can_EnableControllerInterrupts do not modify the software flags. */ /* (CAN208) The function Can_EnableControllerInterrupts shall perform no action when Can_DisableControllerInterrupts has not been called before. */ if (((sint8)0 == Can_ControllerStatuses[Controller].s8IntDisableLevel) && ((uint8)0x1U == u8DisCalledBefore)) { SchM_Exit_Can_CAN_EXCLUSIVE_AREA_01(); /* Processing.....*/

Table continues on the next page...

ID	Subtype	Headline and Description
		} else { SchM_Exit_Can_CAN_EXCLUSIVE_AREA_01(); }
ENGR00347929	NewWork	[CAN] FD follow up -remove magic numbers, optimize code NewWork Description: FD follow up. Create defines and replace magic numbers from code Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00347828	Defect	[CAN] Fix misra error Problem detailed description (how to reproduce it): There are errors in misra report. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: No misra errors was reported. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00339935	NewWork	[CAN] Implements CAN FD support according to Autosar 4.2.1 specification Autosar 4.2.1 was recently released on the market and it includes requirements related to the CAN FD protocol. Also, almost all new hardware platforms released by Freescale(Matterhorn Racerunner, Cobra55, Rayleigh, etc) includes at least one controller(Flexcan, Mcan) which provide support for CAN FD protocol. Based also on customer feedback(e.g. Elektrobit) which asked also to support CAN FD, it is needed that the CAN driver to be updated in order to support the new hardware feature related to CAN FD. In that sense it is needed to create a requirement. A generic proposal for the requirement is: CAN driver shall support CAN FD as an optional feature. The implementation shall be based on the Autosar 4.2.1 specification and on the CAN FD protocol

Table continues on the next page...

ID	Subtype	Headline and Description
		specification.
ENGR00349916	Defect	[CAN] Make checks in functions based on the number of message buffers on each controller Problem detailed description (how to reproduce it): use the maximum message buffers according to resource 96,64, when comparing values. This values will be extracted from resource files and will be stored into an array. For example if the first controller has 96 MB and the second controller has 64 MB then 96 will be saved on first position in array and 64 on the second position. In the IP code the no of MB will be selected using the number of controller (array[controller], array[0]=96) 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): [...]
ENGR00346819	NewWork	[CAN] Remove CER from the code NewWork Description: Remove CER from the code Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00347176	NewWork	[CAN] Remove parameter Can Warning Int Enable Flexcan hardware module allow to generate an interrupt when the tx/rx error counters are bigger than 96. Currently, the generation of this kind of interrupt is enabled using the configuration parameter:Can Warning Int Enable (MCR[WRN_EN]) Current implementation does not clear the interrupt flags related to this kind of interrupt and does not call any user defined callback in order to inform the upper software layers that such kind of event was generated. This behavior is useless. This feature was introduced for the first time into Mcal3.0 code base using cr engr122872. Anyway the cr does not justify the need for this new feature introduction.

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposed solution: due to the missing cPRT requirement and/or customer use case it is better to remove this feature from the code base in order to simplify the code maintainability. Test in: Can_TC_0020, Can_TC_0024, Can_TC_0061
ENGR00350581	Defect	[CAN] Remove some redundant variable Problem detailed description (how to reproduce it): There are some variables does not use for any special purpose. Please remove them for simplify the code. 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): [...]
ENGR00344836	Defect	[CAN] -Correct MISRA errors Problem detailed description (how to reproduce it): Correct MISRA errors 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): [...]
ENGR00335736	NewWork	[CAN] -Disable memory ECC feature NewWork Description: Accroding to TWG decision(ENGR00270740) memory faults shall not be detected/managed by CAN driver. According to that approach CAN driver shall disable this feature during initialization for platforms which includes this feature. Expected behavior: Memory faults detected by ECC shall not be detected/managed by CAN driver. 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): During CAN driver initialization it is needed to assert ECCDIS field from MECR register. This operation shall be done only for platforms which includes in the resource file a parameter related to this feature.
ENGR00338561	Defect	[CAN] -When multiple read periods are used the FIFO is not checked for new message Problem detailed description (how to reproduce it): When more than one period is used for pooling reads, the hardware objects are linked to a specific "pool". For each pool there is defined a specific main function according to CAN442 which in fact call internal function Can_FlexCan_MainFunctionMultipleReadPoll. This function does not check/ cover the situation when a hardware object handle, which use FIFO as reception mechanism, is linked to a reading pool(see the first HRH for each controller). As a consequence the messages from the HRH which use FIFO are not read. Preconditions: FIFO enabled. Multiple read periods are used/defined. Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Packets received by a HRH which use FIFO are not read when multiple read periods are used. 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): Update function Can_FlexCan_MainFunctionMultipleReadPoll from Can_Flexcan.c in order to identify the case when FIFO is used by a specific HRH and then call internal function Can_FlexCan_ProcessRx using as parameters the MB range numbers which cover the interrupt flags related to FIFO(5->7 because 5-frame available, 6-fifo warning, 7-fifo overflow).
ENGR00338535	Defect	[CAN] Function pointers related to FIFO errors can be called even they are NULL Problem detailed description (how to reproduce it): Two callback functions are defined at configuration time in order to be used to announce upper software layers that a FIFO related event(warning, overflow) was generated. At configuration time the callback names can be defined as NULL_PTR even the FIFO support is enabled. At runtime the callbacks are called from Can_FlexCan_ProcessRx without to check if the function pointers(Can_RxFifoOverflowNotification, Can_RxFifoWarningNotification) used to define the callbacks are not null. Call to a NULL function pointer generate unexpected behavior(IVOR)

Table continues on the next page...

ID	Subtype	Headline and Description
		Preconditions: FIFO support enabled 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): Update Can_Flexcan.c in order to check first if the function pointers are not null before to used them.
ENGR00348272	Defect	[FR] Fix WindRiver compiler warnings Problem detailed description (how to reproduce it): WindRiver compiler reports warnings when -Xlint option is used The following errors are reported: - warning: (dcc:1604) "Useless assignment to variable. Assigned value not used" - warning: (dcc:1643) "narrowing or signed-to-unsigned type conversion found: unsigned int to unsigned short" - warning (dcc:1516) "parameter ctrlIdx is never used" Preconditions: - Windriver compiler with -Xlint option set Test Case ID (internal TC that caught the defect) - optional -N/A Trigger: Compile with -Xlint compiler option Observed behavior: - There are warnings see above. Expected behavior: - There are no 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): [...]
ENGR00334620	Defect	[FR] Fix code review findings Problem detailed description (how to reproduce it): During the code rview see ENGR00333846 the following violations were found and have to be fixed - SMCALRule 2.22 -Are the preprocessor define names for sized values following the naming convention? [Rule 2.22] - <MSN|IP>_<VALUE_NAME>_<U|S><8|16|32> - SMCALRule 2.26 - The editor shall be programmed to wrap lines after 250th column. Rationale: to be able to have the page printed and displayed in every system. - SMCALRule 2.35 - Unary operators (++, --, &, ~,!,*), as well as the structure reference operators (., ->) always shall stick to the variable or expression. - SMCALRule 2.31- Pairs of matching parentheses should be located in the

Table continues on the next page...

ID	Subtype	Headline and Description
		same line or in the same column. See the example from SMCAL Coding Guideline. 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): [...]
ENGR00329025	Defect	[FR]Compiler warnings in Fr module Problem detailed description (how to reproduce it): I found the following warning with WindRiver (compiled with option -Xlint): {noformat} Modules: C:\Projects\CheckOut\ACP-7.1\Tresos\plugins \Fr_TS_T2D22M10I1R0\src\Fr_Fripi_LLD.c, 2234: warning (dcc:1772): possible redundant expression, TestSequences: Fr_ComTest_1 Modules: C:\Projects\CheckOut\ACP-7.1\Tresos\plugins \Fr_TS_T2D22M10I1R0\src\Fr_Fripi_LLD.c, 2266: warning (dcc:1772): possible redundant expression, TestSequences: Fr_ComTest_1 Modules: C:\Projects\CheckOut\ACP-7.1\Tresos\plugins \Fr_TS_T2D22M10I1R0\src\Fr_Fripi_LLD.c, 2297: warning (dcc:1772): possible redundant expression, TestSequences: Fr_ComTest_1 Modules: C:\Projects\CheckOut\ACP-7.1\Tresos\plugins \Fr_TS_T2D22M10I1R0\src\Fr_Fripi_LLD.c, 2327: warning (dcc:1772): possible redundant expression, TestSequences: Fr_ComTest_1 Modules: C:\Projects\CheckOut\ACP-7.1\Tresos\plugins \Fr_TS_T2D22M10I1R0\src\Fr_Fripi_LLD.c, 5400: warning (dcc:1772): possible redundant expression, TestSequences: Fr_ComTest_1 {noformat} The warnings are caused by the following expression: ctrlIdx = ctrlIdx; Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Remove the redundant expression: ctrlIdx = ctrlIdx;
ENGR00344560	Defect	[CAN] The Rx FIFO Overflow flag does not clear

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): When a Rx FIFO overflow event occur and CanRxFifoOverflowNotification is different NULL, the driver will report error by calling Can_RxFifoOverflowNotification. Because the flag was not clear properly, this function was call many times. 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): Move the existing line of code which clear the interrupt source flag to a place in order to clear the flag also for the scenarios related to Rxfifo events.
ENGR00331892	Defect	[LIN] Wrong error response when Fault injection is inserted Problem detailed description (how to reproduce it): First, the switch code with LIN_TIMEOUT_ERROR is the same the switch code LIN_BIT_ERROR. That make after compile step, if case LIN_TIMEOUT_ERROR:, it can jump to case LIN_BIT_ERROR instead of (that was I did when run test). More information for this case, I saw ENGR00325422- (Fix compiler warning) did change the LIN_TIMEOUT_ERROR code, that made it the same case LIN_BIT_ERROR. The second, I think the code for case LIN_TIMEOUT_ERROR is not correct. If the LIN_TIMEOUT_ERROR occurs, It should return LIN_RX_NO_RESPONSE instead of LIN_RX_ERROR. My test to cover this CR failed in case of no response (In case of no response, the LIN driver should return LIN_RX_NO_RESPONSE but now it returns LIN_RX_ERROR). 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): [...]
ENGR00330737	NewWork	[MCU] Add support for MC_ME_PCTLnbit NewWork Description:

Table continues on the next page...

ID	Subtype	Headline and Description
		Add support for MC_ME_PCTLn[DBG_F] bit Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00344154	Defect	[MCU] MCU can't change mode when MC_ME_CADDRbit is set in RUN mode Problem detailed description (how to reproduce it): MCU can't change mode when MC_ME_CADDR[RMC] bit is set. 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): [...]
ENGR00350641	Defect	[MCU] Mcu_MC_ME_CheckResetOnModeChange shouldn't be under #if (MCU_CONFIGURE_CADDRN == STD_ON) Problem detailed description (how to reproduce it): Mcu_MC_ME_CheckResetOnModeChange shouldn't be under #if (MCU_CONFIGURE_CADDRN == STD_ON) 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): [...]
ENGR00338107	Defect	[MCU] Remove ERR_IPV_MC_0047 and update PCS addresses Problem detailed description (how to reproduce it):

Table continues on the next page...

ID	Subtype	Headline and Description
		Remove ERR_IPV_MC_0047 and update PCS addresses PCS_DIVE and PCS_DIVS addresses must be updated according to the manual 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): [...]
ENGR00352269	NewWork	[MCU] Review and implement changes according to RM for Halo QM RTM1.0.0 NewWork Description: Review RM version: MAC57D54H_RM_Rev3, 3/2015 http://compass.freescale.net/go/233763719
ENGR00331372	NewWork	[MCU] Update AC9_SC address and add support for writing CLOCKOUT select bits NewWork Description: Update MC_CGM_AC9_SC_ADDR32 with the correct one for Racerunner. Add code to support writing the values configured for CLOCK_OUT0 and CLOCK_OUT1 in the same register. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00338570	Defect	[MCU] Wrong condition in precompile switch in Reg_eSys_MC_RGM.h Problem detailed description (how to reproduce it): Replace: #elif (IPV_MC_05_00_00_10) with #elif (IPV_MC == IPV_MC_05_00_00_10) and #elif (IPV_MC_05_00_00_07) with: #elif (IPV_MC == IPV_MC_05_00_00_07) Preconditions:

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] 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): [...]
ENGR00333528	NewWork	[MCU] Code review against checklist NewWork Description: Review code against code checklist template and correct findings. The initial and final code checklist filled-in files shall be attached to the CR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00330921	Defect	[MCU] Correct compiler warnings Problem detailed description (how to reproduce it): Reg_eSys_MC_RGM.h", line 279: warning #47-D: incompatible redefinition of macro "MC_RGM_DES_F_HSM_DEST_MASK32" (declared at line 205) #define MC_RGM_DES_F_HSM_DEST_MASK32 ((uint32)0x00008000U) 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): [...]
ENGR00335205	NewWork	[MCU] Improve the Mcu_MC_RGM_GetResetReason/ Mcu_MC_RGM_GetResetRawValue function

Table continues on the next page...

ID	Subtype	Headline and Description
		NewWork Description: The Mcu_MC_RGM_GetResetReason/Mcu_MC_RGM_GetResetRawValue functions are improved to the return value dose not depend on platform. Note: - The values of Mcu_ResetType enum must be increased correspondingly with the significant bit of RGM_DES, RGM_FES registers. The first value is MCU_POWER_ON_RESET. - The values of MC_RGM_RAWRESET define must be increased correspondingly with the significant bit of RGM_FES, RGM_DES registers. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00345344	NewWork	[MCU] Save FES and DES in global variable. The customer noticed that the new implementation of Mcu_GetResetReason and Mcu_GetResetRawValue in version 0.9.0 of Calypso MCAL does not save the reset registers in a global variable like it used to be for Bolero or previous version for Calypso. There is no such requirement in Autosar, but is this a wanted change? This implies a redesign of the upper layers for them. Can we update MCU to support this requirement (save the reset registers in a global variable)?
ENGR00332773	NewWork	[MCU] Update the number of the IP version NewWork Classification: (internal task, improvement, feature request) Update IP version and source code with latest version of RM for Panther CUT2.1. 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...)
ENGR00334068	NewWork	[GPT] Review the code against code checklist NewWork Description: Review the code against code checklist. In case of issues found, those should be resolved via peer review follow-up CRs. Please attach the filled-in checklist to this CR before closing it. Code checklist available at (also attached): http://compass.freescale.net/livelink/livelink?

Table continues on the next page...

ID	Subtype	Headline and Description
		func=ll&objId=224108405&objAction=browse Tip: To help automatize checking of some coding rules, you can use AUTO_TOOLS Review Assist tool (BLN_REVIEW_ASSIST_01.00.00). But please be aware this tool is not yet qualified, so all the rules need to be checked by human review.
ENGR00354484	NewWork	[GPT] Review the code against code checklist NewWork Description: Review the code against code checklist. Implement the findings. Please attach the filled-in checklist to this CR before closing it. Code checklist available at (also attached): http://compass.freescale.net/livelink/livelink? func=ll&objId=224108405&objAction=browse Tip: To help automatize checking of some coding rules, you can use AUTO_TOOLS Review Assist tool (BLN_REVIEW_ASSIST_01.00.00). But please be aware this tool is not yet qualified, so all the rules need to be checked by human review.
ENGR00339239	NewWork	[MCU] Add support for version 0x01086100UL NewWork Description: Add support for version 0x01086100UL Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00337721	Defect	[MCU] Loss Of Lock interrupt not available for version 0x02000401UL Problem detailed description (how to reproduce it): Loss of lock interrupt is not available for version 0x02000401UL 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00347400	Defect	[MCU] PLLDIG_PLL1CR_RWBITS_MASK32 is wrong Problem detailed description (how to reproduce it): PLLDIG_PLL1CR_RWBITS_MASK32 is wrong for the version present on Cobra55 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): [...]
ENGR00339516	NewWork	[MCU] Define a sub IP version to distinguish between PLLDIGs NewWork Description: MPC574XR, MPC577XM and MPC574XM have the same IP version of the PLLDIG (02.00.04.13). But the content of the PLLDIG_PLLxCR are different. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00341112	Defect	[MCU] Replace with spaces tabulator character in the source files Problem detailed description (how to reproduce it): There are two tabulator characters in Mcu_PllDig_Pll_LossOfLock_ISR. 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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00334356	Defect	[MCU] RDCR_MEM_SLEEP_EN and RDCR_PAD_KEEP_EN position are incorrect The MEM SLEEP EN bit and PAD KEEP EN bit are defined incorrectly in Reg_eSys_PMCDIG.h #define PMCDIG_RDCR_MEM_SLEEP_EN_MASK32 ((uint32)0x00010000U) #define PMCDIG_RDCR_PAD_KEEP_EN_MASK32 ((uint32)0x00001000U) According to the reference manual MPC5748G Rev 3 , 04/2014 they should be: #define PMCDIG_RDCR_MEM_SLEEP_EN_MASK32 ((uint32)0x00100000U) #define PMCDIG_RDCR_PAD_KEEP_EN_MASK32 ((uint32)0x00010000U)
ENGR00339432	NewWork	[MCU] Add support for IPV_MC 0x04001502UL NewWork Description: Add support for magic carpet version 0x04001502UL Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00353910	NewWork	[GPT] GPT not possible to enable/disable interrupt for RTC Initial Description: In the GPT module the RTC interrupts are always enabled. It is not possible to enable/disable them by calling Gpt_EnableNotification/Gpt_DisableNotification. It seems to be a design choice, but why? The customer would like to use the enable/disable feature to avoid wasting resources when it's not necessary. Observed behavior: Gpt_EnableNotification/Gpt_DisableNotification cannot enable/disable interrupt for RTC timer When can it be observed? (at configuration time, at runtime, at compile time?) Runtime Expected behavior: Gpt_EnableNotification/Gpt_DisableNotification cannot enable/disable interrupt for RTC timer Reported release baseline: BLN_SMCAL_4.0_CALYPSO_BETA_0.9.0 Proposed solution (Optional):
ENGR00347801	NewWork	[ADC] Add new resource support normal trigger for Adcdig NewWork Description: Add new resource support normal trigger for Adcdig Expected behavior: [...]

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): [...]
ENGR00354161	NewWork	[DIO] Magic number in Dio_Siul2_ReverseBits16 The customer checked the source code and found the magic number is used in the following lines in Dio_Siul2.c: 294, 296, 298, 300, 301. It could be replaced by macros for maintenance ability improvement.
ENGR00328588	NewWork	[DIO] Update new structure in resource files NewWork Description: Update new structure in resource files Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00354222	NewWork	[ICU] Review the code against code checklist NewWork Description: Review the code against code checklist. Implement the findings. Please attach the filled-in checklist to this CR before closing it. Code checklist available at (also attached): http://compass.freescale.net/livelink/livelink?func=ll&objId=224108405&objAction=browse Tip: To help automatize checking of some coding rules, you can use AUTO_TOOLS Review Assist tool (BLN_REVIEW_ASSIST_01.00.00). But please be aware this tool is not yet qualified, so all the rules need to be checked by human review.
ENGR00350106	NewWork	[ICU] Update Misra violations Problem detailed description (how to reproduce it): Misra violations Preconditions: [...] 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: Generate misra report. Proposed solution (Optional): Fix Misra violations
ENGR00344114	Defect	[MCU] MCAL contains M4 macros (in pdf and src files) A grep for 'M4_' over all modules shows following files which still contains M4 macros: [Can] AUTOSAR_MCAL_CAN_UM.pdf: The ASR version was not replaced thus all links in the document are not working [Lin] AUTOSAR_MCAL_LIN_UM.pdf: There are still many M4 macros in the document. [Mcu] Mcu_Reg_eSys_SIUL2.h: Macro M4_SRC_FAMILY is not replaced [Mcu] Reg_eSys_MC_ME.h: Macros M4_SRC_MODUE_NAME and M4_SRC_AR_MODUE_DEPENDENCY are not replaced.
ENGR00341737	Defect	[PORT] Port output function is not refresh correctly The bit set/clear macro is used to configure ODC but this field consist of three bits. PORT_Siul2_SetPinDirection/PORT_Siul2_RefreshPortDirection use bit set and bit clear macro to configure ODC. but ODC has 3bits, the macro will not work for some case
ENGR00333712	Defect	[MCU] Identifier "SSCM_UOPS_ADDR32" is undefined Problem detailed description (how to reproduce it): Identifier "SSCM_UOPS_ADDR32" is undefined. 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): [...]
ENGR00336125	NewWork	[GPT] Analyze data flow anomalies from the StaticAnalysis report NewWork Description: Analyze and fix all problems for the data flow anomalies in the static analysis report. All suspicious variables usage that are not errors need to be documented in the Remarks column.

Table continues on the next page...

ID	Subtype	Headline and Description
		More details on data flow anomalies can be found in the attachement
ENGR00346067	Defect	[GPT] Update Misra warning Problem detailed description (how to reproduce it): Update misra warning 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): [...]
ENGR00344470	NewWork	[GPT] Update share interrupt NewWork Description: IPV_STM using share interrupt for ISR_1 - ISR_3 Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00343931	Defect	[GPT] Missing #define GPT_START_SEC_CODE in Gpt_Stm.c Problem detailed description (how to reproduce it): There is GPT_STOP_SEC_CODE but no GPT_START_SEC_CODE in Gpt_Stm.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):

Table continues on the next page...

ID	Subtype	Headline and Description
		[...]
ENGR00337999	NewWork	[WDG] Comment MISRA warning 8.10 NewWork Description: Comment MISRA warning 8.10 in WDG_Swt_Isr.c Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00345288	Defect	[ICU] The interrupt flag should be cleared after clearing the interrupt enable bit. Problem detailed description (how to reproduce it): In the Icu driver, the flag is cleared then the interrupt enable bit is cleared. This is not good since the flag is set right after cleared it is finally not cleared. Icu_SIUL2_DisableInterrupt The Icu_WKPU_DisableInterrupt: LOCAL_INLINE FUNC (void, ICU_CODE) Icu_WKPU_DisableInterrupt(VAR(uint32, AUTOMATIC) u32channelMask) { SchM_Enter_Icu_ICU_EXCLUSIVE_AREA_20(); { /* Clear ISR Flag */ /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_WRITE32(WKPU_WISR, u32channelMask); /* Disable IRQ Interrupt */ /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_BIT_CLEAR32(WKPU_IRER, u32channelMask); /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_BIT_CLEAR32(WKPU_WRER, u32channelMask); } SchM_Exit_Icu_ICU_EXCLUSIVE_AREA_20(); } Should be LOCAL_INLINE FUNC (void, ICU_CODE) Icu_WKPU_DisableInterrupt(VAR(uint32, AUTOMATIC) u32channelMask) { SchM_Enter_Icu_ICU_EXCLUSIVE_AREA_20(); { /* Disable IRQ Interrupt */ /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_BIT_CLEAR32(WKPU_IRER, u32channelMask); /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_BIT_CLEAR32(WKPU_WRER, u32channelMask);

Table continues on the next page...

ID	Subtype	Headline and Description
		/* Clear ISR Flag */ /* @violates @ref Icu_WKPU_c_3 Conversions shall not be performed between a pointer to a function and any type other than an integer type */ REG_WRITE32(WKPU_WISR, u32channelMask); } SchM_Exit_Icu_ICU_EXCLUSIVE_AREA_20(); }
ENGR00351484	NewWork	[ICU] Update the single interrupt handling scheme NewWork Description: Single interrupts (1 ISR for all WKPU sources) should be handled in a diferent WKPU ISR, that is adequately named Expected behavior: WKPU single interrupt handling to be improved. Requirement source: - (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Create a new ISR for handling such cases and not use WKPU_EXT_IRQ_0_7_ISR like before
ENGR00336334	Defect	[LIN] Correct the name of register bit mask Problem detailed description (how to reproduce it): In Lin_NonASR.c, we have some register bit masks, which were changed via code review checklist, but these weren't updated in this file. 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): [...]
ENGR00346710	NewWork	[LIN] Empty Note field in the External Assumptions IM chapter NewWork Description: Some of the drivers have in their Integration Manuals - Chapter: External Assumptions, notes stating irrelevant information, e.g.: - 'Ok' (for ADC) / - 'SM', 'No able to check it.' (for LIN) - 'SM' (for SPI) Expected behavior: The irrelevant comments for IM's Notes field shall be updated in DOORS

Table continues on the next page...

ID	Subtype	Headline and Description
		module and regenerated into IMs.
ENGR00335927	Defect	[LIN] Module cannot compile with Dem errors configured Problem detailed description (how to reproduce it): Module cannot compile with Dem errors 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): [...]
ENGR00348669	Defect	[LIN] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED,

Table continues on the next page...

ID	Subtype	Headline and Description
		CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00328162	Defect	[LIN] WUF bit in LINSR register is set when entering in DRUN mode Problem detailed description (how to reproduce it): After the initializations of the clocks, the WUF bit in the LINSR register is set. This is blocking the LIN driver to be initialized because, due to the implementation of the driver, before the initialization, the wake-up is checked and it will be detected. The driver will not be initialized. A Siebel ticket is raised: SR# 1-2297930404 : FSLFAE: [CALYPSO][LINFLEX] WUF bit in LINSR register is set when entering in DRUN mode. 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): [...]
ENGR00336850	Defect	[LIN] Wrong usage of Dem functionality Problem detailed description (how to reproduce it): The DEM functionality is reversed. In the plugin, when "LinDisableDemReportErrorStatus" is checked/enabled, the DEM functionality must be disabled. In the current implementation, the define is generated with STD_ON. The correct generation is STD_OFF. And all the usage of this functionality, in the code, must be made if the define is STD_OFF Proposed solution: Add a note in the plugin that says "If the checkbox is set, the DEM functionality is DISABLED" All statements that have #if (LIN_DISABLE_DEM_REPORT_ERROR_STATUS == STD_ON) Should be replaced with #if (LIN_DISABLE_DEM_REPORT_ERROR_STATUS == STD_OFF)
ENGR00328709	NewWork	[MCL] Add SOC dependent compiler definition NewWork Description: Following compiler switches were removed from Base,they must be added to MCL: #define IPV_DMA #define DMA_NB_CONTROLLERS #define DMA_NB_CHANNELS #define DMAMUX_NB_MODULES #define DMAMUX_NB_CHANNELS Expected behavior:

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): [...]
ENGR00337304	Defect	[MCL] Dma error notification for the Dma Instance 0 is not editable Problem detailed description (how to reproduce it): Dma error notification is not editable in Tresos when "Mcl Dma Error NotificationSupported" is checked. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: Dma error notification is not editable when "Mcl Dma Error NotificationSupported" is checked. Expected behavior: Dma error notification is not editable when "Mcl Dma Error NotificationSupported" is checked. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update xdm to make it editable when Mcl Dma Error NotificationSupported = true
ENGR00354968	NewWork	[MCL] Fix MISRA issues in Calypso RTM 1.0.0 relase NewWork Description: Fix Misra issues Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00336429	Defect	[MCL] Follow-up: fix the code checklist findings Problem detailed description (how to reproduce it): Fix the code checklist findings: update variable names,function names and preprocessor defines names in order to comply to the project coding style rules. For more details, the findings are available also in the attachment of the peer review. Preconditions: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]
ENGR00332312	Defect	[MCL] Generate the symbolic name for DMA channel There are some APIs provided by MCL library for DMA configuration like void Mcl_DmaEnableHwRequest(Mcl_ChannelType ChannelNumber), Mcl_DmaEnableHwRequest however there is no sysmblic name generated for Mcl channel, the customer dont know what should be passed for the argument ChannelNumber. Please generate the symbolic name or index to use with the public APIs. Please also update user manual for this usage.
ENGR00349676	NewWork	[MCL] Make um/im more generic NewWork Description: For all the platform specific stuff use the variables_common.xml variables. In this way, the manuals will always be automatically generated with the correct values, no manually update will be needed. Please see Fee example. Eg of update: -platform name -sw versions -tresos id etc Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00330695	NewWork	[MCL] Remove all production error reporting to SERR and use only DEM NewWork Description: All production error reporting shall be done using the AutoSar defined mechanism - DEM. All calls to SerrNotifyError shall be replace by DemReportErrorEvent. Proposal for implementation of DEM reporting is in the attachment. For details on how to implement Dem handling please see the attachement Expected behavior:

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): [...]
ENGR00344651	NewWork	[MCL] Support enable/disable initialization DCHMID register NewWork Description: Update Resource and config files for support enable/disable initialization DCHMID register Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00331154	Defect	[MCL] Support the polling mode for MclErrorChecking Problem detailed description (how to reproduce it): Current implementation, the customer can choose polling or interrupt mode in EB Tresos configuration plugin but for the polling mode, there is no function to be polled by upper layer. The polling function should be provided by Mcl driver to be called from upper layer. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: [...] Expected behavior: iTWG (19.09.2014): - ticket ENGR00331601 Agreed by iTWG:cPRT to be update with the following req.: "The MCL module shall provide two APIs for error status checking the for global errors (for all channels) and for a single channel." Proposed solution (Optional): [...]
ENGR00340381	NewWork	[MCL] Support the req CPR-MCAL-732.mcl for platform that have 2 DMA NewWork Description: Support the req CPR-MCAL-732.mcl for platforms that have 2 DMA instances. Expected behavior: [...] Requirement source:

Table continues on the next page...

ID	Subtype	Headline and Description
		cPRD (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Support the req CPR-MCAL-732.mcl for platforms that have 2 DMA instances.
ENGR00334156	Defect	[MCL] Trigger capability is supported for logical channels 0-4 Problem detailed description (how to reproduce it): In the xdm configuration the trigger enable(MclDMAChannelTriggerEnable) is availabale for logical channels 0-4. It should be available for the first DMAMUX hardware channels, as specified by the hardware manual. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: In the xdm configuration the trigger enable is availabale for logical channels 0-4 Expected behavior: It should be available for the first 4 hardware channels from each DMAMUX Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00348670	Defect	[MCL] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16

Table continues on the next page...

ID	Subtype	Headline and Description
		ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/

Table continues on the next page...

ID	Subtype	Headline and Description
		ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00344786	NewWork	[MCL]Update the address of DMAMUX channel configuration registers NewWork Description: - DMAMUX in some platform (ex: Rayleigh) that based on ARM cpu don't using the Little Endian for access memory. So need to update source code and resource files for check the Little or Big Endian. A new resource parameter will be added to specify the access type for DMAMUX channel configuration registers. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00346097	Defect	[MCL] Register DMA_DCHMID is not configured accordingly Problem detailed description (how to reproduce it): Register DCHMID is not configured according to the options set in Tresos. The default values will be used. The reason is the fact that the register initialization is guarded by IPV_DMA define which is undefined. 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 usage of IPV_DMA and IPV_DMA_02_00_00_00 from MCL.There is a resource parameter to guard DCHMID, so the defines are not needed any more.
ENGR00347153	NewWork	[MCL] Update interrupt handler for IPV_DMA NewWork Description: In some platforms (ex: Matterhorn, Rayleigh ...) have only one type of interrupt for DMA, they are one vector interrupt for one DMA channel or one vector interrupt for all of channels. But on Cobra55 have both of them, on DMA_A of Cobra55 have one vector for one channel, but on DMA_B have one vector for all of channel of DMA_B. So need update the interrupt handler to support 3

Table continues on the next page...

ID	Subtype	Headline and Description
		case: 1. One vector - One DMA channel 2. One vector - Multi DMA channel 3. Both 1, 2 Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00349125	Defect	[MCU] "BAF Disable" parameter is not configured in configuration file. Configure "BAF Disable" parameter with different value but the same source code generated in the configuration file.
ENGR00350255	NewWork	[MCU] Add an additional check when generating CADDR parameters NewWork Description: Add an additional check when generating CADDR parameters in order to have consistency between EPD and XDM tests. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00350505	NewWork	[MCU] Add support for Fault Injection Test point NewWork Description: Add support in MCAL drivers for Fault Injection Test points. Details in attached presentation and on page: http://zcz01web02:1080/wiki/vav/Wiki%20Pages/TestDesignWithFaultInjection.aspx Expected behavior: Updated drivers to add fault injection points Requirement source: N/A (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): -Add in Base the following defines: PUT_IN_QUOTES, MCAL_FAULT_INJECTION_POINT -> they should be empty in case fault injection tests are not compiled (using define ?d ENABLE_FAULT_INJECTION) -In test files (tickets for ITG) In the fault injection tests make files we need to have ?d ENABLE_FAULT_INJECTION

Table continues on the next page...

ID	Subtype	Headline and Description
		-In driver code (tickets for developers) Add label in driver code (tester and developer should agree on the label name and the place in the code) MCAL_FAULT_INJECTION_POINT (LABEL_NAME); Note: LABEL_NAME should be compliant to the following naming rule: <MDL>_FIP_<COUNT>_<COMMENT> Eg: FLS_FIP_1_UPDATE_VARIABLE_A
ENGR00344491	Defect	[MCU] Add support to set the boot address before enable multi-core Problem detailed description (how to reproduce it): We are facing the emulation debug port fail error in Mcu_init(); The below execution flow shows the place from which emulation debug port fail has occurred. Mcu_init()----> Mcu_IPW_Init()----> Mcu_MC_ME_Modes_Periph_Init()----> Mcu_MC_ME_ApplyMode()----> REG_WRITE32(MC_ME_MCTL_ADDR32, ((uint32)ModeConfig | (uint32)MC_ME_MCTL_INVERTED_KEY_U32)); After writing inverted key (0x0000A50F) to the control register MC_ME_MCTL , emulation debug port fail has occured and controller get reset. 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): [...]
ENGR00341156	Defect	[MCU] CGM_AUX_CLOCK1 to be remloved from EPD and from xdm files Apparently Aux CLOCK1 has been removed from the RM, it doesn't exist anymore. So it should be removed from the xdm config file, and from the EPD file? In tresos it seems that it cannot be edited anyway, but the customers are using their own tool managed to configure it, causing compiler issues
ENGR00335982	Defect	[MCU] Correct some MISRA errors Problem detailed description (how to reproduce it): Error(s) Mcu_Cfg.h", line 238, MISRA Rule Violated 19.7 (Advisory): no MISRA violation comment was found (maybe wrong format is used). Mcu_Cfg.h", line 238, MISRA Rule Violated 19.4 (Required): no MISRA violation comment was found (maybe wrong format is used).

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]
ENGR00330854	NewWork	[MCU] FMEA reports shall contain a link between safety measures and requirement IDs All FMEA shall contain the link between the safety measure and the requirement ID or the external assumption generated due to that safety measure. A new column shall be added in our current FMEAs to indicate what is the DOORS ID and the text of the requirement. For more details see the attachement..
ENGR00346916	Defect	[MCU] MCU_START_SEC_CODE is redefinition in Mcu.c Problem detailed description (how to reproduce it): MCU_START_SEC_CODE is defined two times for the same section in Mcu.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): [...]
ENGR00337915	NewWork	[MCU] Mcu_MC_ME_ModeConfigType should be renamed to be applicable on all platforms Problem detailed description (how to reproduce it): There is an impact on the implementation of Mcu_ConfigType structure between MPC574XP and MAC58RXXX. Mcu_MC_ME_ModeConfigType is not available on MAC58RXXX. Preconditions: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		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): Rename "Mcu_MC_ME_ModeConfigType" to "Mcu_ModeConfigType"
ENGR00329320	Defect	[MCU] Mcu_SscmGetUops requirement not fully implemented Problem detailed description (how to reproduce it): The PR-MCAL-3286.mcu is not fully implemented. A variable must be introduced in the plugin, and generate the define MCU_SSCM_GET_UOPS_API In Mcu_IPW.c, a function must be added to call Mcu_SSCM_GetUops(). This function must be guarded by the define above mentioned. Please see the implementation from other platforms. 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): [...]
ENGR00330696	NewWork	[MCU] Remove all production error reporting to SERR and use only DEM NewWork Description: All production error reporting shall be done using the AutoSar defined mechanism - DEM. All calls to SerrNotifyError shall be replace by DemReportErrorEvent. Proposal for implementation of DEM reporting is in the attachment. For details on how to implement Dem handling please see the attachement 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
ENGR00348248	Defect	[MCU] The MVRON is readonly bit in MC_ME_STANDBY_MC and writable in STOP0 Problem detailed description (how to reproduce it): The MVRON bit is readonly bit it is written with 1. Please see the attached configuration 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): [...]
ENGR00354278	Defect	[Mcu] Fix Misra errors Misra rule running reports errors. Proposed solution: Fixes Misra rule errors for Calypso
ENGR00354990	Defect	[PORT] Fix error misra in Port_Cfg.h Problem detailed description (how to reproduce it): Fix error misra in Port_Cfg.h 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): [...]
ENGR00340779	NewWork	[PORT] Introduce support for pins that are defined in RM as input and output to work also as input/output Introduce support for pins that are defined in RM as input and output to work also as input/output in the same time.

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00329466	Defect	[PORT] PK[4] (PCR 164) has wrong emios channels for MPC574XC Problem detailed description (how to reproduce it): The port configuration for PK4 offers the wrong emios channels to be selected from. While it is correct for MPC57xG devices, it is not correct for MPC574xC devices: PK[4] 164 0000_0000 GPIO[164] SIUL2 PK[4] 0000_0001 CAN5TX FlexCAN_5 PK[4] 0000_0010 LIN8TX LIN_8 PK[4] 0000_0011 E1UC_1_H EMIOS1 PK[4] 0000_0100 E0UC_9_H EMIOS0 PK[4] 549 0000_0101 E1UC_1_H EMIOS1 PK[4] 521 0000_0011 E0UC_9_H EMIOS0 Tresos offers to select between emios1 and emios2, but emios0 and emios1 is correct (see above from signal table from *C devices). 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): [...]
ENGR00350489	Defect	[PORT] SMC bit in MSCR is set only in case of GPIO Problem detailed description (how to reproduce it): The customer wants to set the PortPinSafeMode option for all kinds of output. Our current implementation does it only for pins configured as GPIO output. Preconditions: see above. Observed behavior: see above When can it be observed? (at configuration time, at runtime, at compile time?) at configuration time. Expected behavior: PortPinSafeMode option for all kinds of output. Proposed solution (Optional): [...]
ENGR00325647	Defect	[PORT] Wrong description of PortPinPcr Problem detailed description (how to reproduce it): The PortPinPcr is described as Pin Id of the port pin.This value will be assigned to the symbolic name derived from the port pin container short name. This is for PortPinId not for PortPinPcr.

Table continues on the next page...

ID	Subtype	Headline and Description
		PortPinPcr is used to specify port configuration register: SIUL I/O Pin Multiplexed Signal Configuration Registers (MSCR number). This is not correct, please update a correct description for PortPinPcr. 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): [...]
ENGR00329470	Defect	[PORT] Wrong port direction for Adc Mux channels Problem detailed description (how to reproduce it): The Tresos validation forces to configure all adc channels to be inputs. However, e.g. ADC0_MA[2] on PCR2 is an output channel. 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): [...]
ENGR00344799	NewWork	[PORT] setting incorrect direction for GPIO function Problem detailed description (how to reproduce it): On EB, when I configure a pad with GPIO function. The value of MSCR generated incorrect for direction. 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

Table continues on the next page...

ID	Subtype	Headline and Description
		source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00337267	NewWork	[PWM] Configuration of 16bit/24bit timer should be easier to handle NewWork Description: For some applications run-time is critical (16bits may be favored) but for other precision is more important (thus 24bit would be used) Currently the only way to select if a timer is on 16bit or 24bit is by changing the predefined values in resource files. Expected behavior: However this means that from customer point of view they can not chose to use 16bit or 24bit for a platfrom that supporst both this options and would make sense to provide customer this option if is avaialble. Proposed solution (Optional): For platforms that support both 16bit and 24bit timers the selection between the types of timers should be configurable. This means that: For eMios platfroms in the resource files, the field: Pwm.Emios16BitsVariant should be changed to: Pwm.TimerPrecision: PWM_16BIT, PWM_24BIT Same parameter should be used for GTM platfroms. For platfroms like Calypso that support only 16bit emios: the 24bit option should be removed from resource files. In XDM file a new parameter: PwmTimerPrecision should be added in Pwm/PwmGeneral/ This parameter should be of type: REFERENCE, and should reference the resource value: Pwm.TimerPrecision: base on the value of this define: - for emios based plaforms the following define should be generated in Pwm_Cfg.h #define PWM_EMIOS_16_BITS_VARIANT STD_ON -> if the value of PwmTimerPrecision = PWM_16BIT else #define PWM_EMIOS_16_BITS_VARIANT STD_OFF - for gtm based plaforms the following define should be generated in Pwm_Cfg.h #define PWM_GTM_16_BITS_VARIANT STD_ON -> if the value of PwmTimerPrecision = PWM_16BIT else #define PWM_GTM_16_BITS_VARIANT STD_OFF
ENGR00333780	NewWork	[PWM] Implement delta PR-MCAL-3326 requirements Align to CPRD 1.23 so folowing requirements need to be implemented: PR-MCAL-3326: The Pwm driver shall provide optional API and functionality for switching selected PWM outputs to software control. It shall utilize DTSRCSEL register. Per default this optional API and functionality shall be disabled. Details of the implementation for each of the requirements is described by the following CRs PR-MCAL-3326: ENGR218380
ENGR00328247	NewWork	[PWM] Implement Pwm_SetCounterBus functionality (CPR-MCAL-729)

Table continues on the next page...

ID	Subtype	Headline and Description
		Implement Pwm_SetCounterBus functionality (CPR-MCAL-729)
ENGR00336356	Defect	[PWM] Max default period The max default period of PWM channel in Matterhorn Beta 0.9.0 is 65534. It is 16-bit value, It could be 65535 for 16 bit value and 16777215 for 24bit value.
ENGR00332293	NewWork	[PWM] PWM ATOM channels support 24bits counter and compare Initial Description: PWM ATOM channel support 24 bits counter/compare. However the driver uses Pwm_PeriodType is 16bits therefore the customer cannot get the 24 bit resolution and max period for PWM ATOM channel. Will this 24 bit Pwm_PeriodType be supported? (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...)
ENGR00329293	Defect	[PWM] PwmClockRef should be selected for each eMios module Problem detailed description (how to reproduce it): Currently, ony one PwmClockRef is used for all eMios in Pwm. However, the eMios clock could be different from each eMios depending on the Global Prescaler. Preconditions: [...]

Table continues on the next page...

ID	Subtype	Headline and Description
		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): Add PwmClockReferencePoint for PWM (similar GPT driver), then move PwmClockRef parameter into PwmeMios container for each module. (The customer can select CUSTOM source then specify the frequency in MCU module for McuClockReferencePoint)
ENGR00350087	Defect	[PWM] Pwm_SetTriggerDelay will raise a DET error when call with channel is configured invalid mode Problem detailed description (how to reproduce it): The Pwm_SetTriggerDelay should raise a DET error when call with channel is configured invalid mode Proposed solution (Optional): In HLD layer, should check the mode of current channel, if mode is different the OPWMT, a DET error should be raised
ENGR00334814	NewWork	[PWM] Remove the Dem error const variable name Cloned from Gpt. Same logic, as below, applies to PWM also. Initial Description: If we configure Gpt_E_Forbidden_Invocation in GptDemEventParameterRefs, Gpt_Cfg.h, Gpt_Cfg.c and Gpt_PBcfg.c generate following lines. Gpt_Cfg.h: {code} extern CONST(Dem_EventIdType, GPT_CONST)Gpt_E_Forbidden_Invocation; {code} Gpt_Cfg.c: {code} CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} Gpt_PBcfg.c: {code} CONST(Dem_EventIdType, GPT_CONST) Gpt_E_Forbidden_Invocation = (Dem_EventIdType)DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation; {code} However, Gpt_E_Forbidden_Invocation will be defined like below in our Dem_IntErrId.h: {code} #define DemConf_DemEventParameter_Gpt_E_Forbidden_Invocation 27U #define Gpt_E_Forbidden_Invocation 27U

Table continues on the next page...

ID	Subtype	Headline and Description
		#define Dem_Gpt_E_Forbidden_Invocation 27U {code} Our EB Dem module can be used in all ASR versions and therefore is backward compatible to older ASR versions: * 1st line: for ASR 4.0.3 and up * 2nd line: for ASR 3.1 * 3rd line: for ASR 3.2 and up to ASR 4.0.2 So, because of this, the above results in like these. {code} extern CONST(Dem_EventIdType, GPT_CONST)27U; CONST(Dem_EventIdType, GPT_CONST) 27U = (Dem_EventIdType)27U; {code} And due to this, compile of Gpt_PBcfg.c fails. Problem detailed description (how to reproduce it): compile of Gpt_PBcfg.c fails. Preconditions: [...] Observed behavior: Compiler error When can it be observed? (at configuration time, at runtime, at compile time?) at compile time Expected behavior: No compiler error Reported release baseline: MPC577XM 0.9.0 Proposed solution (Optional): To prevent such name clashes a pre/post-fix could be added to this const in the code in a future release.
ENGR00332168	NewWork	[PWM] Review and update User manual and Integration Manual NewWork Description: Please review in all manuals the following items: - Interrupt service routine and interrupt vector are documented correctly. - The non-autosar/ Freescale specific parameters are documented with a correct description. - Remove unavailable parameters/APIs in the platform. - Deviation from ASR. Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00340459	Defect	[PWM] The notification function still extern when the PwmNotification is FALSE Problem detailed description (how to reproduce it): The notification function still extern when the PwmNotification is FALSE Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional

Table continues on the next page...

ID	Subtype	Headline and Description
		[...] 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): [...]
ENGR00345506	NewWork	[PWM] The period in master bus is not suitable NewWork Description: In pwm channel, use can be select the period default in tick or not. But in Master bus channel, user only select period in tick, so it is not suitable Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00332419	Defect	[PWM] Update IM MCL dependency for ADC, SPI when use in DMA mode and PWM, GPT, ICU Problem detailed description (how to reproduce it): Please add into Chapter 5: Mcl should be init before ADC, SPI in DMA mode. MCL should be called before ICU, PWM, GPT (ATOM used). Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00321603	Defect	[PWM] Update User Manual for Pwm_GetOutputState API Problem detailed description (how to reproduce it): The user manual stated that Pwm_GetOutputState API is not fully implemented and in 3.4 Runtime Errors stated that: Pwm_GetOutputState Pwm_E_Service_Unavailable Service Pwm_GetOutputState is called. There is no support in the IPs used by PWM on this platform for getting the output state of a Pwm channel. The service reports this error and always

Table continues on the next page...

ID	Subtype	Headline and Description
		returns PWM_LOW value However,in the API code, The Pwm_GetOutputState returns the output state read from internal register! This is inconsistent with UM. Please update UM to reflect this functionality! 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): [...]
ENGR00347258	Defect	[PWM]Internal counter bus should not used with DAOC mode Problem detailed description (how to reproduce it): Follow RM of Rainier Rev 4, the internal counter bus is not used on DAOC mode, but now user can be select DAOC mode with internal counter bus. This defect only impact to Rainier platform. 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): [...]
ENGR00354781	NewWork	[PWM]fix misra violation NewWork Description: Det has changed, so some misra violation occur Expected behavior: Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): add comment rules 16.10
ENGR00347181	Defect	[SPI] Add SPI limitation for Spi_AsyncTransmit without DMA

Table continues on the next page...

ID	Subtype	Headline and Description
		Problem detailed description (how to reproduce it): Following limitation should be stated in the User Manual. When DMA is not used, Spi_AsyncTransmit uses EOQ to update TxFiFo and get data from RxFiFo. Due to the hardware specification, when EOQ is set for every 4 frames, the CS will be inactive regardless of CONT bit. Therefore, when the user configure the Job in the continuous mode, the CS will be inactive then active again after every 4 frames.
ENGR00347846	NewWork	[SPI] Add a better description for SPI driver with DCACHE ON on the manual NewWork Description: The SPI driver that uses DMA, when DCACHE is ON has some dependencies. In order to use the driver with the DMA when DCACHE = ON, the user must follow the following things: 1. do not use internal bufers or 2. put the variables from SPI_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE in a RAM zone where it is NON CACHEABLE. Please add these specifications in the manual at the section "Data Cache Restriction". Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00335634	Defect	[SPI] Check the valid range for the parameter HWUnit in the function Spi_SetHWUnitAsyncMode Problem detailed description (how to reproduce it): In file Spi.c, function Spi_SetHWUnitAsyncMode, if user use the invalid value (greater than total number hardware unit) for the parameter "HWUnit", driver will fail to run. 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): [...]

Table continues on the next page...

ID	Subtype	Headline and Description
ENGR00339300	Defect	[SPI] Configuration parameter SpiChannelType is not acording to Autosar Specification Problem detailed description (how to reproduce it): Configuration Parameter "SpiChannelType" is not according to the AUTOSAR ECU Configuration Parameter Definition (AUTOSAR_MOD_ECUConfigurationParameters.xml), respectively you can also check SPI206_Conf: <SHORT-NAME>EB</SHORT-NAME> respectively <SHORT-NAME>IB</SHORT-NAME> Instead the current implementation of SPI is using SPI_EB respectively SPI_IB. Preconditions: N/A Test Case ID (internal TC that caught the defect) - optional [...] Trigger: N/A Observed behavior: Configuration Parameter SpiChannelType not acording Autosar ECUC. Expected behavior: Configuration Range/Value of Spi_ChannelType as per Autosar ECUC. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00345734	NewWork	[SPI] Configuration parameter for period time of Spi_MainFunction call The customer reports that Configuration parameter for period time of Spi_MainFunction call is missing: In AUTOSAR 4.0.x the BSW modules description should contain the _mainFunction calls of the BSW modules. For generation of this description the period time of the _mainFunction call is needed. This period time should be defined in a configuration parameter (as it is done for Can module eg CanMainFunctionWritePeriod). For Spi module this config parameter is still not defined in AUTOSAR (we raised an ASR bugzilla ticket: RfC 55718). In the mean time the vendor should define a vendor specific parameter therefore. Please consider this in the next delivery.
ENGR00336319	Defect	[SPI] Correct API ID Problem detailed description (how to reproduce it): Using wrong API ID in line 2056 of Spi.c file. else if (HWUnit >= SPI_MAX_HWUNIT) { /* Call Det_ReportError */ Det_ReportError((uint16) SPI_MODULE_ID, (uint8) 0, SPI_GETHWUNITSTATUS_ID_U8, SPI_E_PARAM_UNIT_U8); Status = (Std_ReturnType)E_NOT_OK;

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]
ENGR00331546	Defect	[SPI] Correct GPIO selection case in template files Problem detailed description (how to reproduce it): In current code: [!IF "SpiEnableCs = 'true'"!] [!IF "SpiCsSelection = 'CS_VIA_PERIPHERAL_ENGINE'"!] (uint32)(DSPI_CMD_[!"SpiCsIdentifier"!]_U32 | /* Chip Select Pin */ [!ELSEIF "SpiCsSelection = 'CS_VIA_GPIO'"!][!// [!ENDIF!] [!IF "SpiCsContinous = 'FALSE'"!][!WS "8"!] DSPI_CMD_CONTINOUS_FALSE_U32[!// [!ELSEIF "SpiCsContinous = 'TRUE'"!][!WS "8"!] DSPI_CMD_CONTINOUS_TRUE_U32[!// [!ELSE!][!ERROR "Illegal value for SpiCsContinous"!][!// [!ENDIF!]), /* Continous chip select */ [!ELSE!][!WS "8"!]0x00000000u, [!ENDIF!] [!IF "SpiCsPolarity = 'HIGH'"!][!// [!WS "8"!] DSPI_[!"SpiCsIdentifier"!]_IDLEHIGH_U32[!// [!ELSE!][!// [!WS "8"!](uint32)0u[!// [!ENDIF!][!WS "2"!]/* Chip select polarity */ Should be modified below: [!IF "SpiEnableCs = 'true'"!] [!IF "SpiCsSelection = 'CS_VIA_PERIPHERAL_ENGINE'"!][!// (uint32)(DSPI_CMD_[!"SpiCsIdentifier"!]_U32 | /* Chip Select Pin Via Peripheral Engine*/ [!IF "SpiCsContinous = 'FALSE'"!][!WS "12"!] DSPI_CMD_CONTINOUS_FALSE_U32),/* Continuous chip select */ [!ELSEIF "SpiCsContinous = 'TRUE'"!][!WS "12"!] DSPI_CMD_CONTINOUS_TRUE_U32),/* Continuous chip select */ [!ELSE!][!ERROR "Illegal value for SpiCsContinous"!][!// [!ENDIF!][!// [!ELSEIF "SpiCsSelection = 'CS_VIA_GPIO'"!][!// 0x00000000u,/* Chip Select Via GPIO */ [!ENDIF!][!// [!ELSE!][!WS "8"!]0x00000000u, [!ENDIF!] [!IF "SpiCsPolarity = 'HIGH'"!][!// [!WS "8"!] DSPI_[!"SpiCsIdentifier"!]_IDLEHIGH_U32[!// [!ELSE!][!WS "8"!](uint32)0u[!//

Table continues on the next page...

ID	Subtype	Headline and Description
		[!ENDIF!][!WS "2"!]/* Chip select polarity */ 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): [...]
ENGR00346711	NewWork	[SPI] Empty Note field in the External Assumptions IM chapter NewWork Description: Some of the drivers have in their Integration Manuals - Chapter: External Assumptions, notes stating irrelevant information, e.g.: - 'Ok' (for ADC) / - 'SM', 'No able to check it.' (for LIN) - 'SM' (for SPI) Expected behavior: The irrelevant comments for IM's Notes field shall be updated in DOORS module and regenerated into IMs.
ENGR00337450	Defect	[SPI] Fix compiler warning Problem detailed description (how to reproduce it): We have the warning "constant out of range" in Spi.c file line 773 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): [...]
ENGR00334820	Defect	[SPI] Fix misra errors Problem detailed description (how to reproduce it): With the latest version of Misra tool, there were some errors discovered. Please correct them. See attached log for the errors.

Table continues on the next page...

ID	Subtype	Headline and Description
		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): [...]
ENGR00326221	Defect	[SPI] GPIO Chip Select Signal is not triggered/cannot be triggered Problem detailed description (how to reproduce it): According to the description of the configuration parameter SpiCsSelection, the user must implement the job start and end notifications to trigger the Chip Select in case SpiCsSelection is set to CS_VIA_GPIO. Using synchronous jobs (SpiHwUnitSynchronous=SYNCHRONOUS), the notifications aren't called. Since the SPI driver does trigger the CS or any job notification there is no possibility to trigger CS signal at all. According to AUTOSAR specification the SPI driver must - trigger the CS by itself, even if SpiCsSelection is set to CS_VIA_GPIO When does this happen: ---------- During runtime and anytime the SPI driver is used to send or receive data. In which configuration does this happen: ---------- SpiCsSelection=CS_VIA_GPIO and hardware unit is set to SYNCHRONOUS. Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00355081	Defect	[SPI] Incorrect CS management of the SPI driver What happens (symptoms): ---------- The CS has a wrong state between the channels (channel count > 1) in one job! (see attached picture SPI_Trace.png) When does this happen: ---------- During runtime. In which configuration does this happen: ----------

Table continues on the next page...

ID	Subtype	Headline and Description
		When continous mode for the CS (switched by periphal engine!) of the SPI driver will be used. Discussion: ---------- Please find some information attached (generated source files, Spi configuration, Trace picture). Issue was found on Spi MPC574XG MCAL4_0_BETA_0_9_0 incl. HF4
ENGR00338071	Defect	[SPI] Missing the delimitter {} in the if-condition case Problem detailed description (how to reproduce it): In Spi.c, function Spi_Init(), if we enable the det error detection inline 832, we missed the { in the end of case. 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): [...]
ENGR00335954	Defect	[SPI] Module cannot compile with Dem errors configured Problem detailed description (how to reproduce it): Module cannot compile with Dem errors 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): [...]
ENGR00329297	Defect	[SPI] SpiClockRef should select clock reference for each hardware module Problem detailed description (how to reproduce it): The SPI baudrate clock source are different depending on the hardware module. DSPI_2, DSPI_3: F80

Table continues on the next page...

ID	Subtype	Headline and Description
		DSPI_0, DSPI_1: F40 SPI_0: F40 SPI_1, SPI_2 : F80 SPI_3, SPI_4, SPI_5 :F40 However, there is only one SpiClockRef used for all hardware module. The actual bit rate will be different from configured value for certern module. 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): Add SpiClockReferencePoint for Spi (similar GPT driver), then move SpiClockRef parameter into each SpiPhyUnit
ENGR00347623	Defect	[SPI] The baudrate for classic mode is up to 40Mhz in MPC574xG Problem detailed description (how to reproduce it): According to MPC574xG datasheet (attached) in the Low Speed Mode, the t_sck is up to 20Mhz (50ns). In High Speed Mode, the t_sck is up to 40Mhz (20ns). e.g: If DSPI2 has (DSPI Baud Clock F80-Max 80Mhz Clock) and PBR=0 -> Divide to 2, BR=0-> divide to 2, DBR=0, we will have Spi_clock = 80Mhz/ (2*2) = 20Mhz, this satisfies the condition min period (50ns). For the DSPI[3:5], DSPI0:1], SPI[0], the DSPI Baud Clock is F40 (40Mhz), the DBR should be 1 to support 20Mhz. However the baudrate is limited to 16Mhz under EB Tresos. (attached customer configuration). When 20Mhz is configured, it throws an error: description: Value "2.0E7" of node "/AUTOSAR/TOP-LEVEL-PACKAGES/Spi/ELEMENTS/Spi/SpiDriver/SpiDriver_0/SpiExternalDevice/SpiExternalDevice_0/SpiBaudrate" is out of range: For input string: "The value is out of range. In normal SPI mode the value must be in the range 4..16000000Hz. In the TSB mode the value must be in the range 4..40000000Hz" Preconditions: TSB is not enabled. Observed behavior: Error message throwed by EB Tresos Expected behavior: No Error message
ENGR00342727	Defect	[SPI] The end notification function will not be called when use GPIO chip select Problem detailed description (how to reproduce it): When we call Spi_SyncTransmit() with (SPI_OPTIMIZE_ONE_JOB_SEQUENCES == STD_ON), then call

Table continues on the next page...

ID	Subtype	Headline and Description
		Spi_Ipw_SyncTransmit_Fast(), the end notification function will not be called Otherwise, when we call Spi_Ipw_SyncTransmit() and have one job failed, all remaining jobs will be terminated, but the remaining end notification functions will not be called 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): [...]
ENGR00332417	Defect	[SPI] Update IM MCL dependency for ADC, SPI when use in DMA mode and PWM, GPT, ICU Problem detailed description (how to reproduce it): Please add into Chapter 5: Mcl should be init before ADC, SPI in DMA mode. MCL should be called before ICU, PWM, GPT (ATOM used). Preconditions: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): [...]
ENGR00348674	Defect	[SPI] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED

Table continues on the next page...

ID	Subtype	Headline and Description
		ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described. Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32

Table continues on the next page...

ID	Subtype	Headline and Description
		Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE ICU_START_SEC_VAR_INIT_32/ICU_STOP_SEC_VAR_I
ENGR00345055	Defect	[SPI] Wrong Spi baudrate generation Problem detailed description (how to reproduce it): MPC574xG MCAL 4.0 0.9.0 HF2 Customer found an issue with the generation of the baudrate when several SPI units are used (possibly with different source clocks). The generated CTAR register is not correct, the calculated baudrate is not the expected one. After reproducing the issue I found that the problem is in the code generation template. For instance in Spi_Cfg.c template, line 803 must be replaced by [! SELECT "node:ref(../../../../SpiGeneral/SpiPhyUnit/*[number($varSpiHwUnit)]/ SpiPhyUnitClockRef)"!][!// (and the same in the PB or LT versions). The current implementation will always select the first item from the list, so always the same clock value, which ends up in a wrong calculation of the baudrate. 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): [...]
ENGR00336853	Defect	[SPI] Wrong usage of Dem functionality Problem detailed description (how to reproduce it): The DEM functionality is reversed. In the plugin, when "SpiDisableDemReportErrorStatus" is checked/enabled, the DEM functionality must be disabled. In the current implementation, the define is generated with STD_ON. The correct generation is STD_OFF. And all the usage of this functionality, in the code, must be made if the define is STD_OFF Proposed solution:

Table continues on the next page...

ID	Subtype	Headline and Description
		Add a note in the plugin that says "If the checkbox is set, the DEM functionality is DISABLED" All statements that have #if (SPI_DISABLE_DEM_REPORT_ERROR_STATUS == STD_ON) Should be replaced with #if (SPI_DISABLE_DEM_REPORT_ERROR_STATUS == STD_OFF)
ENGR00353855	Defect	[WDG] Compiler error since no WDGIF_TYPES_AR_RELEASE_XXX in the customer's WDGIF module Problem detailed description (how to reproduce it): AutoSar Version Numbers of Wdg_Cfg.h and WdgIf_Types.h are different: The problem appears because Wdg driver requests the version of WdgIf_Types, which is not defined. From what I've noticed so far the AUTOSAR standard specifies that only the version of WdgIf module shall be defined Observed behavior: Compiler error Expected behavior: No Compiler error Proposed solution (Optional): CE's comment: The inter-module version check is not necessary to be done in the header file. The WDGIF_TYPES_AR_RELEASE_XXX are not available. The check of these macro should be removed.
ENGR00351092	NewWork	[WDG] Create driver FMEA for Calypso 6M RTM1.0.0 NewWork Description: Create driver FMEA for Calypso 6M RTM1.0.0
ENGR00329269	Defect	[WDG] DEM should be optional Problem detailed description (how to reproduce it): The DEM reference container is optional parameter (0..1) but now it is 1..1. e.g In Wdg driver: 1) In the EB Tresos configuration plugin, according to the ASR specs the multiplicity of WdgDemEventParameterRefs container, WDG_E_DISABLE_REJECTED, WDG_E_MODE_FAILED is 0..1 but it is now 1..1. 2) Dem_EventIdType now is used in Wdg_Cfg.h and Wdg_CfgExt.c Therefore at the configuration time, DEM is always required under EB Tresos. At the compile time, DEM is now wrapped by SERR, however, the Dem_EventIdType is used, and there are some wrong check in configuration files (Wdg_43_Instance0_LCfg.c-Wdg_43_Instance2_LCfg.c, Wdg_43_Instance0_PBCfg.c-Wdg_43_Instance2_PBCfg.c). #ifndef DISABLE_MCAL_INTERMODULE_ASR_CHECK /* Check if source file and Dem header file are of the same Autosar version */ #if ((WDG_AR_RELEASE_MAJOR_VERSION_PBCFG_C != DEM_AR_RELEASE_MAJOR_VERSION) || \ (WDG_AR_RELEASE_MINOR_VERSION_PBCFG_C != DEM_AR_RELEASE_MINOR_VERSION)) #error "AutoSar Version Numbers of Wdg_43_Instance0_PBCfg.c and Dem.h are different" #endif

Table continues on the next page...

ID	Subtype	Headline and Description
		#endif DEM may not be used and it is not included indirectly here. The version check should be done for serr only. But Serr_NotifyErrorEvent is not used in these files, so this check could be eliminated. 3) The integration manual should be updated. SERR is stub sample code that wraps Dem module, this should be replaced by the actual code. 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): [...]
ENGR00336587	Defect	[WDG] Follow-up: fix the code checklist findings Problem detailed description (how to reproduce it): Fix the code checklist findings:update variable names and preprocessor defines names in order to comply to the project coding style rules. For more details, the findings are available also in the attachment of the peer review. Preconditions: [...] Test Case ID (internal TC that caught the defect) - optional [...] Trigger: [...] Observed behavior: The variables and preprocessing define name does not comply to the project naming rules. Expected behavior: The variables and preprocessing define name should comply to the project naming rules. Note: in the "Expected behavior" field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Update variable names and preprocessor defines names in order to comply to the project coding style rules.
ENGR00330698	NewWork	[WDG] Remove all production error reporting to SERR and use only DEM NewWork Description: All production error reporting shall be done using the AutoSar defined mechanism - DEM. All calls to SerrNotifyError shall be replace by DemReportErrorEvent.

Table continues on the next page...

ID	Subtype	Headline and Description
		Proposal for implementation of DEM reporting is in the attachment. For details on how to implement Dem handling please see the attachement Expected behavior: [...] Requirement source: [...] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00348675	Defect	[WDG] Update Sections to be defined in MemMap.h in Integration Manual Problem detailed description (how to reproduce it): The chapter "Sections to be defined in MemMap.h" description is inconsistent among MCAL drivers. It it better to have the consistent format, typo in all drivers. The Adc driver Integration Manual does not have the list of all memory sections, it provides only the formulation while the other drivers provides the details about all memory sections. Adc: NO_INIT_UNSPECIFIED_NO_CACHEABLE section is not described. The IM should provides the all sections in details like Fee driver's Integration Manual. The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: ADC_START_SEC_CONFIG_DATA_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_INIT_UNSPECIFIED ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ADC_START_SEC_CODE/ADC_STOP_SEC_CODE ADC_START_SEC_VAR_NO_INIT_8/ADC_STOP_SEC_VAR_NO_INIT_8 ADC_START_SEC_VAR_NO_INIT_32/ADC_STOP_SEC_VAR_NO_INIT_32 ADC_START_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE/ ADC_STOP_SEC_VAR_NO_INIT_UNSPECIFIED_NO_CACHEABLE ADC_START_SEC_CONST_32/ADC_STOP_SEC_CONST_32 ADC_START_SEC_CONST_16/ADC_STOP_SEC_CONST_16 ADC_START_SEC_CONST_8/ADC_STOP_SEC_CONST_8 Can Driver: The driver should provide the correct information about all memory sections. e.g: On matterhorn MCAL 1.0.0: CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED, CAN_START_SEC_VAR_INIT_UNSPECIFIED/ CAN_STOP_SEC_VAR_INIT_UNSPECIFIED are not described. While SEC_VAR_UNSPECIFIED_NO_CACHEABLE does not exist in the driver. Dio Driver: The driver should provide the correct information about the memory sections. e.g: On matterhorn MCAL 1.0.0: Only DIO_START_SEC_CONFIG_DATA_UNSPECIFIED/ DIO_STOP_SEC_CONFIG_DATA_UNSPECIFIED and DIO_START_SEC_CODE/DIO_STOP_SEC_CODE are valid. The others does not exist in Dio driver. Eth Driver: ETH_START_SEC_VAR_NO_INIT_8/ ETH_STOP_SEC_VAR_NO_INIT_8 is not described.

ID	Subtype	Headline and Description
		Fls Driver: Many section are not valid. e.g For matterhorn MCAL RTM1.0.0, only following section are valid FLS_START_SEC_CONFIG_DATA_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_CODE/FLS_STOP_SEC_CODE FLS_START_SEC_CODE_AC/FLS_STOP_SEC_CODE_AC FLS_START_SEC_CONFIG_DATA_8/FLS_STOP_SEC_CONFIG_DATA_8 FLS_START_SEC_CONST_UNSPECIFIED/ FLS_STOP_SEC_CONFIG_DATA_UNSPECIFIED FLS_START_SEC_VAR_INIT_UNSPECIFIED/ FLS_STOP_SEC_VAR_INIT_UNSPECIFIED FLS_START_SEC_VAR_INIT_8/FLS_STOP_SEC_VAR_INIT_8 FLS_START_SEC_VAR_INIT_32/FLS_STOP_SEC_VAR_INIT_32 FLS_START_SEC_VAR_INIT_BOOLEAN/ FLS_STOP_SEC_VAR_INIT_BOOLEAN FLS_START_SEC_CONST_32/FLS_STOP_SEC_CONST_32 Fr Driver: Gpt Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. GPT_START_SEC_CONFIG_DATA_UNSPECIFIED/ GPT_STOP_SEC_CONFIG_DATA_UNSPECIFIED GPT_START_SEC_CODE/GPT_STOP_SEC_CODE GPT_START_SEC_VAR_NO_INIT_32/GPT_STOP_SEC_VAR_NO_INIT_32 GPT_START_SEC_CONST_32/GPT_STOP_SEC_CONST_32 GPT_START_SEC_VAR_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_INIT_UNSPECIFIED GPT_START_SEC_VAR_NO_INIT_UNSPECIFIED/ GPT_STOP_SEC_VAR_NO_INIT_UNSPECIFIED GPT_START_SEC_VAR_INIT_32/GPT_STOP_SEC_VAR_INIT_32 Icu Driver: The Integration Manual should provide the more details about all memory sections like Dio/Fr driver's Integration Manual. e.g: On matterhorn MCAL 1.0.0, the following section should be described. ICU_START_SEC_CODE/ICU_STOP_SEC_CODE ICU_START_SEC_CONFIG_DATA_UNSPECIFIED/ ICU_STOP_SEC_CONFIG_DATA_UNSPECIFIED ICU_START_SEC_VAR_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_UNSPECIFIED/ ICU_STOP_SEC_VAR_NO_INIT_UNSPECIFIED ICU_START_SEC_VAR_NO_INIT_32_NO_CACHEABLE/ ICU_STOP_SEC_VAR_NO_INIT_32_NO_CACHEABLE

4.7 BETA 0.9.0

This is the first release for the MPC574XG AUTOSAR 4.0 MCAL from the new source base, it replaces all previous MPC574XG AUTOSAR 4.0 MCAL releases.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number Release Notes for MPC574XG
AUTOSAR 4.0 MCAL RTM 1.0.5
Revision 1.0