Changes List for Freescale Semiconductor, Inc. AUTOSAR MPC563xM MCAL3.0 swv: 2.0.0 RTM

==========

What's Modified v2.0.0

==========

ID	Headline	Description
ENGR00140170	IPV_FLASH] Implement robustness code review improvement proposals for Multiple returns	Implement the findings of the robustness code review for multiple returns found in the following functions (this point remained unresolved from ENGR00138402 updates). Please refer to the attached document, which contains the updated comments for all the Findings under the column "Other comments". 1) Fls_LLD_MainFunction( ) 2) Fls_LLD_VerifyErase( ) 3) Fls_LLD_VerifyWrite( ) 4) Fls_LLD_SectorErase( ) 5) Fls_LLD_SectorWrite( ) 6) Fls_LLD_SectorCompare( )
ENGR00153060	[ADC] Adc_ReadGroup optimization	NewWork Description: Adc_ReadGroup optimization Expected behavior: Based on actual implementation a typical reda sequence is going trough the following steps: Adc_Init(); Adc_SetupResultBuffer(GroupNumber, &ResultBuffer); Adc_StartGroupConversion(GroupNumber); While(ADC_State < BUSY){} Adc_ReadGroup(GroupNumber, &UserBuffer); ….. An optimized version of the Adc_ReaGroup could avoid to loop on all the channel inside the group and copying the result directly inside the expected result buffer. Then the last step in the previous sequence can be cahnged passing directly the user destination buffer Adc_ReadGroup(GroupNumber, &ResultBuffer); This optimization can be applied only if the stream depth is limited to 1. Requirement source: Customer Request
ENGR00155506	[ADC] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD

ENGR00170279	[ADC] Change AdcChannelSampTime from ENUMERATION to INTEGER	Problem detailed description (how to reproduce it): VSMD check reports the error that the AdcChannelSampTime xdm field is ENUMERATION instead of INTEGER (like standard AUTOSAR). Preconditions: N/A Trigger: N/A Observed behavior: VSMD report with error. When can it be observed? (at configuration time, at runtime, at compile time?) configuration time Expected behavior: VSMD report 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): Change the field type.
ENGR00142205	[ADC] Code Coverage analysis don't reach the specified threshold (> 90%)	Code Coverage analysis don't reach the specified threshold (> 90%)
ENGR00160680	[ADC] Cosmetic Adc driver improvements	Several cosmetic Adc driver improvements, for all platforms, can be done. Please see the analysis tab.
ENGR00163332	[ADC] Development for Monaco 1.5M RTM 2.0.0	Development for Monaco 1.5M RTM 2.0.0
ENGR00151849	[ADC] ISR optimization	NewWork Description: Optimization for ISR Expected behavior: Inside the ISR there is a loop to allign each sample. It is requested to make this loop optional and controlled by a configuration checkbox. The default value shall be alignement enabled (to guarntee backward compatibility Requirement source: Customer

ENGR00162683	[ADC] Incorrect memory mapping	Problem detailed description (how to reproduce it): Please check if all drivers' symbols are placed in between START/STOP sections present on MemMap.h file. Usually these sections are the following: drv_START_CONFIG_DATA_UNSPECIFIED drv_STOP_CONFIG_DATA_UNSPECIFIED drv_START_SEC_CODE drv_STOP_SEC_CODE drv_START_SEC_VAR_UNSPECIFIED drv_STOP_SEC_VAR_UNSPECIFIED drv_START_SEC_CONST_UNSPECIFIED drv_STOP_SEC_CONST_UNSPECIFIED The following findings were raised by our clients. Even if the current driver is not recorded in the list below please check the consistency. > The following symbols are missing a correct memory mapping. All of these symbols get assigned to section ".text_vle": > MCAL (Leopard BETA 0.9.1, but also RTM1.0.0) > * Functions: > * CAN_BASE_ADDRS: Not found at all > * DIO_SIUL_ReadChannel(), > DIO_SIUL_WriteChannel(), > DIO_SIUL_ReadPort(), > DIO_SIUL_WritePort(), > DIO_SIUL_ReadChannelGroup(), > DIO_SIUL_WriteChannelGroup(): > + Dio_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Dio_TS_T2D17M0I9R0\include\Siul_LLD_dio.c: Not mapped > + Port_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Port_TS_T2D17M0I9R0\src\Siul_LLD_port.c: Not mapped
ENGR00170544	[ADC] Remove unused Reg_eSys_EQADC.h inclusion from configuration source files	Problem detailed description (how to reproduce it): Useless Reg_eSys_EQADC.h inclusion into ADC configuration source files. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) compile time Expected behavior: Reg_eSys_EQADC.h can be removed and the driver builds yet Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Remove unused Reg_eSys_EQADC.h inclusion from configuration source files

ENGR00170462	[ADC] Remove unused symbolic link and cfg test folder	Problem detailed description (how to reproduce it): Unsed files are yet in place. Preconditions: [...] Trigger: [...] Observed behavior: Symbolic links not refer any files. When can it be observed? (at configuration time, at runtime, at compile time?) compile time Expected behavior: REmove them Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00163388	[ADC] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.
ENGR00140913	[ALL] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00151943	[ALL] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00153822	[ALL] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD

ENGR00142202	[ALL] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]
ENGR00160015	[ALL] Configuration variable naming in generated code for ADC and WDG	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.
ENGR00142702	[ALL] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]

ENGR00163608	[BASE] Add CONSTP2FUNC define in Mcal.h	In Compiler.h file the definition of P2FUNC is different by what is written in SWS. Why? SWS: /* COMPILER039 */ Example (Metrowerks, S12X): #define P2FUNC(rettype, ptrclass, fctname) rettype (*ptrclass fctname) Example (Cosmic, S12X): #define P2FUNC(rettype, ptrclass, fctname) ptrclass rettype (*fctname) Compiler.h /* COMPILER039 */ #define P2FUNC(rettype, ptrclass, fctname) rettype (*fctname) If the implementation is changed like in SWS, some allocation errors will be corrected.
ENGR00142913	[BASE] Comment modification for DSPI_0	1) The modification in Base – DSPI_0 must be marked in the comments as “not implemented on Monaco, but needed by SPI driver”. 2) The “XPC563XM” is hardcoded in the file. The M4 macro “M4_SRC_USED_PERIPHERAL” must be used to allow M4 replacements during plugin generation
ENGR00162473	[BASE] Extended support for Micro Second Bus	NewWork Description: To support the Micro Second Bus feature implemented for SPI driver, an update is needed in the BASE module for "Reg_eSys.h" file. It is needed to have a macro definition for DSPI_0 unit as shown below. /** @brief DSPI 0 @details M4_SRC_USED_PERIPHERAL System Memory Map @remarks Not implemented on Monaco, but needed by SPI driver */ #define DSPI0_BASEADDR ((uint32)0xFFF90000UL)
ENGR00162830	[BASE] General updates for Monaco RTM 2.0.0	NewWork Description: Development for Monaco RTM2.0.0 […] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00143504	[BASE] In traceability matrix 'sign off' details not updated	In traceability matrix 'sign off' details are not updated (Name, Date).
ENGR00142917	[BASE]-The base address for DSPI_0 shall be added Reg_eSys.h file.	DSPI_0 not implemented on Monaco, but needed by SPI driver. This modification is already fixed in Spi CR 139511.
ENGR00169817	[BASE]Update Mcal.h to support Cosmic Compiler	Modify Mcal.h file to support Cosmic Compiler.
ENGR00160630	[BUILDENV] Add Cosmic Compiler and update the Build Env for Monaco RTM 2.0.0	Add the Cosmic Compiler. Add all features not included in specific or Integration CRs.
ENGR00162933	[BUILD_ENV] Update variables_common.xml file to be used for MONACO UM/IM documentation	NewWork Description: Update "variables_common.xml" file to have FSL specific information Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00151507	[CANIF] Update the stubs, to no longer export any BASE headers	NewWork Description: - Update the stubs, to no longer export any BASE headers Expected behavior: - No BASE headers are visible by including the stubs headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00155499	[CAN] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142206	[CAN] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]
ENGR00160948	[CAN] Configuration variable naming in generated code	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.

ENGR00162926	[CAN] General updates for MONACO RTM 2.0.0	NewWork Description: Update all the files related to below mentioned release activities: AMDC / VSMD reports generation/review MISRA checks UM/IM updating Code Size/Stack Usage reports LDRA Reports generation Traceability matrix Profiling report generation EPD test report Summary reports Review checklist (code, design test plan, requirements) Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00160681	[CAN] Memory mapping	Conformance tests results for the CAN drivers are attached. Please analyze them and fulfill the analysis report. Memory mapping is done in FlexCan_LLD.c and Can_LLD.c, it should be applied to FlexCan_LLD.h and Can_LLD.h.
ENGR00170494	[DEM] DEM timeout reporting in Fls_LLD_Cancel	Add a DEM error report in case of timeout for all three while loops in Fls_LLD_Cancel function.
ENGR00163605	[DEM] New macro for production error required by GPT driver	NewWork Classification: (internal task, improvement, feature request) internal task In case for feature request, does a TWG ticket exist (cloned or linked)? No NewWork Description: We have to add a new definition for "GPT_E_TIMEOUT_TRANSITION" for DEM error. This needs to be defined in Dem_IntErrId.h file. Expected behavior: Dem_IntErrId.h file contains the definition for "GPT_E_TIMEOUT_TRANSITION" . Requirement source: Internal refactoring Proposed solution (Optional): [...]
ENGR00151509	[DET] Update the stubs, to no longer export any BASE headers	NewWork Description: - Update the stubs, to no longer export any BASE headers Expected behavior: - No BASE headers are visible by including the stubs headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00155510	[DIO] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD

ENGR00159997	[DIO] DIO cannot configure GPIO145	Problem detailed description (how to reproduce it): In MCAL 3.0 for Monaco (MPC563xM), version 1.9.0 + HF5 and HF6, the DIO modules reports an error when trying to configure GPIO145. However this pin is vailable on all packages. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00169838	[DIO] DIO development for Monaco - release RTM 2.0.0	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...) Proposed solution (Optional): [...]
ENGR00163395	[DIO] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.
ENGR00151510	[ECUM] Update the stubs, to no longer export any BASE headers	NewWork Description: - Update the stubs, to no longer export any BASE headers Expected behavior: - No BASE headers are visible by including the stubs headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00155512	[FEE] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD

ENGR00142207	[FEE] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [Statement Coverage, Branch/Decision Coverage, MC/DC Profile is not in the 90% for BETA As soon as possible execute LDRA using the updated test suite and check what line are uncovered the test suite accordingly to have 100% of coverage.] Preconditions: [...] 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): [...]
ENGR00162750	[FEE] Integrate FSL robustness tests into the standard MCAL testing (2 part)	NewWork Description: Integrate remaining and updated FSL robustness tests into the standard MCAL testing Expected behavior: FEE050 test FEE050 test 2 write_test write_test_2 swap_test (has been removed) immediate_test run for all compiler. Requirement source: Code and .cmm file must be updated for all compiler Proposed solution (Optional): the attached files should be integrated in MCAL structure. A separationbetween generic files and specific files should be done. Sholud be used different labels in .cmm files and code files to let all compilers to compile Test patterns.
ENGR00162743	[FEE] Test development for Monaco RTM 2.0.0	NewWork Description: [- Update test suite to the llatest modification - also update the test specification] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00163405	[FEE] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.
ENGR00155511	[FLS] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142208	[FLS] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): Statement Coverage 86.63 % Branch/Decision Coverage: 81.11 % MC/DC Profile : 63.75 % As Monaco doesn't have DFO and Interleaved block many lines can be justified, It will be done on RTM As soon as possible execute LDRA using the updated test suite and check what line are uncovered because of interleaved block or DFO which will be justified and update the test suite to cover the remaining. Preconditions: None 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): [...]

ENGR00162428	[FLS] Fls.c doesn't compile as Cer_ReportError is not defined	Issue Type: Bug Issue classification: high(A) Priority: high Package: Fls MPC56XXK Found in version: MCAL3.0_0.9.1_BETA What happens (symptoms): ---------- It is not possible to compile "Fls.c". Compiler cannot resolve used macros (e.g. Cer_ReportError) for "CER". When does this happen: ---------- Always In which configuration does this happen: ---------- Any Discussion: ---------- include of "Mcal.h" is missing in Fls.c or Fls.h.
ENGR00170154	[FLS] Update UM to detail Fls sector start address meaning (logical vs. physical)	The FLS User Manual does not give sufficient information regarding what type of addess the "Fls Sector Start Address" refers to: it gives the impression that it contains the physical address, but it is actually the logical address. This lack of detail creates misunderstanding at the customer side, like shown in the example below. UM needs to be updated. === Customer complaint: === During integration of the latest MCAL (RTM 3.0.0) for Pictus an issue occurs. All start addresses from FLS_CODE_ARRAY_0_A_SHADOW to FLS_DATA_ARRAY_0_TEST are invalid, i.d. no erase or write for these sectors are possible with the Fls driver. Solution: Change sector start addresses from -> to * FLS_CODE_ARRAY_0_A_SHADOW = 0x80000 -> 0x00200000 * FLS_CODE_ARRAY_0_A_TEST = 0x84000 -> 0x00400000 * FLS_DATA_ARRAY_0_L00 = 0x88000 -> 0x00800000 * FLS_DATA_ARRAY_0_L01 = 0x8c000 -> 0x00804000 * FLS_DATA_ARRAY_0_L02 = 0x90000 -> 0x00808000 * FLS_DATA_ARRAY_0_L03 = 0x94000 -> 0x0080C000 * FLS_DATA_ARRAY_0_TEST = 0x98000 -> 0x00C00000
ENGR00163404	[FLS] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.

ENGR00170089	[FLS] test suite development and quality package update for Monaco RTM 2.0.0	NewWork Description: - Development for Monaco RTM 2.0.0. Quality Package: - Generate Stack Size Report - Generate MISRA report - Generate VSMD report - Generate CodeCoverage Report and StaticAnalysis report (LDRA) - Generate Profile report - Generate UM/IM - Generate the Test Specification document Test Report: - Run all tests, generate single tests report and the final SummaryTest Report (.xml and .html files)
ENGR00170349	[FLS] while loops without timeout and DEM reporting in configuration files	1) In the below mentioned while loop in Fls_LLD_Cancel, the timeout is not present. while(!(Fls_LLD_regBasePtr[FLASHMEM_MCR] & MCR_DONE)) { /* wait until end of abort * } 2) Also the DEM error should be present in case of timeout. Both points are valid for all three while loops in Fls_LLD_Cancel function.
ENGR00155501	[GPT] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142209	[GPT] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]

ENGR00162927	[GPT] General updates for MONACO RTM 2.0.0	NewWork Description: Update all the files related to below mentioned release activities: AMDC / VSMD reports generation/review MISRA checks UM/IM updating Code Size/Stack Usage reports LDRA Reports generation Traceability matrix Profiling report generation EPD test report Summary reports Review checklist (code, design test plan, requirements) Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00143507	[GPT] In traceability matrix 'sign off' details not updated	In traceability matrix 'sign off' details are not updated (Name, Date).
ENGR00169703	[GPT]-Update Misra violation for GPT	NewWork Description: fixed misra violation for pitrti Fix misra violation for GPT Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00155500	[ICU] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00160606	[ICU] Delayed reading of duty cycle when using eMios channel In IPWM mode	Found:BLN_MCAL_3.0_BOLERO3M_BETA_0.9.0 May also involve other MCUs with eMIOS Problem detailed description (how to reproduce it): When using an eMIOS unified channel that does in IPWM mode , When the sampling of the values is done before the falling edge(if the active edge is rising)it will return the values recorded from the previous period Attached is a test that reproduce the problem.In this a pwm signal is generated and period and duty cycle is modified. A second timer is used to place the call of Icu_GetDutyCycleValues along the generated signal.This can be seen better in the oscilloscope capture attached. When you adapt the test take care that : -the variable "channel" is set to a icu channel that is configured to a hardware EMIOS UC that supports IPWM mode. - define ICU_TEST_PWM_CHANNEL_1 as an eMIOS unified channel that supports OPWFMB - define ICU_TEST_TIMER_CHANNEL as an eMIOS unified channel that has internal counter - the variable connection is correctly set so that the pwm output is read by the input icu channel For a more detailed description of the problem please see the attached .pptx file.

ENGR00160687	[ICU] Function prototypes to be mapped	Conformance tests results for the ICU drivers are attached. Please analyze them and fulfill the analysis report. In SIUL_Icu_LLD_IRQ.c file local functions are not mapped.
ENGR00162928	[ICU] General updates for MONACO RTM 2.0.0	NewWork Description: Update all the files related to below mentioned release activities: AMDC / VSMD reports generation/review MISRA checks UM/IM updating Code Size/Stack Usage reports LDRA Reports generation Traceability matrix Profiling report generation EPD test report Summary reports Review checklist (code, design test plan, requirements) Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00158759	[ICU] ICU IM contains wrong exclusive area information.	Problem detailed description (how to reproduce it): In file eMIOS_Icu_LLD.c the function "eMIOS_LLD_EnableInterrupt" use the exclusive area 13. However according to AUTOSAR_MCAL_ICU_IM.pdf P28. Chapter5 "eMIOS_LLD_GetInputState" uses exclusive area 13. In file Icu_LLD.c exclusive area 8 is used by function "Icu_LLD_DisableEdgeCount" However according to AUTOSAR_MCAL_ICU_IM.pdf P27 Chapter 5 "eMIOS_LLD_DisableEdgeCount" uses the exclusive area 8 In file Siu_Icu_LLD.c Function "SIU_LLD_EnableInterrupt" uses the exclusive area 17. However according to AUTOSAR_MCAL_ICU_IM.pdf P28 Chapter 5 "SIUL_LLD_DisableInterrupt" uses the exclusive area 17 In file Siu_Icu_LLD.c Function "SIU_LLD_DisableInterrupt" uses the exclusive area 18. However according to AUTOSAR_MCAL_ICU_IM.pdf P28 Chapter 5 "SIUL_LLD_DisableInterrupt" uses the exclusive area 18 Please check for corectness all exclusive areas from the IM. Preconditions: NA Trigger: NA Observed behavior: Wrong information included in the IM When can it be observed? (at configuration time, at runtime, at compile time?) Doc. Expected behavior: Updated IM info Proposed solution (Optional): Update IM info
ENGR00143508	[ICU] In traceability matrix 'sign off' details not updated	In traceability matrix 'sign off' details are not updated (Name, Date).

ENGR00160022	[ICU]- comments or code should not exceed 100 characters	Problem detailed description (how to reproduce it): All the files for Icu driver needs to be check so that it will not exceed 100 characters [...] Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00170419	[ICU]-Improve the code coverage for platform which usew IPV_EMIOS	Improve the code coverage for platform which uses IPV_EMIOS.
ENGR00169729	[ICU]-Update Misra violation for ICU	NewWork Description: fixed misra violation for emios Fix misra violation for Icu_LLD.c Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00159816	[IPV_DMAV2] Driver generates compiler warnings when compiled with DIAB 5.8.0.0	Problem detailed description (how to reproduce it): While compiling with DIAB 5.8.0.0, IPV_DMAV2 driver generates below compiler warnings. "narrowing or signed-to-unsigned type conversion found: " Preconditions: DIAB 5.8.0.0 compiler version shall be used for building driver. Trigger: [...] Observed behavior: Compiler warning is generated during building driver. When can it be observed? (at configuration time, at runtime, at compile time?) compile time 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 the compiler warning in driver files. If it can not be fixed, update the comments in driver code with proper doxygen tags.
ENGR00151461	[IPV_DMAV2]: MISRA Violation comments are missing for some of the violation	NewWork Description: [MISRA Violation comments are missing for some of the violation] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]

ENGR00151668	[IPV_DMAV2]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00156100	[IPV_DMA] Incorrectly generated "Dma_Mcu_LLD.c"	Found: Pictus RTM 1.0.1 In the MCU plugin the file "Dma_Mcu_LLD.c" had to be changed. There is a complex #if statement that prevented the current LIN configuration from working. The problematic statement is commented out, and below is the simplified one: /*#if (((USE_ADC_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==123) && (ADC_USE_DMA_LLD==STD_ON)) || \ ((USE_ADC_MODULE==STD_OFF) && (USE_SPI_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==83) && \ (SPI_USE_DMA_LLD==STD_ON)) || \ ((USE_ADC_MODULE==STD_OFF) && (USE_SPI_MODULE==STD_OFF) && (USE_LIN_MODULE==STD_ON) && \ (DMA_COMMON_MODULE_ID==82) && (LIN_USE_DMA_LLD==STD_ON)) || \ ((USE_ADC_MODULE==STD_OFF) && (USE_SPI_MODULE==STD_OFF) && (USE_LIN_MODULE==STD_OFF) && \ (USE_MCU_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==101) && (MCU_USE_DMA_LLD==STD_ON)))*/ #if (((USE_ADC_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==123) && (ADC_USE_DMA_LLD==STD_ON)) || \ ((USE_SPI_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==83) && (SPI_USE_DMA_LLD==STD_ON)) || \ ((USE_LIN_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==82) && (LIN_USE_DMA_LLD==STD_ON)) || \ ((USE_MCU_MODULE==STD_ON) && (DMA_COMMON_MODULE_ID==101) && (MCU_USE_DMA_LLD==STD_ON))) Because there is no appropriate configuration to get the condition in comments working for LIN module integration it looks like a bug in source code. Attached Lin.xdm
ENGR00140909	[IPV_DMAv2] Fado RTM - Code Review against checklist	Driver code should not cross 100 character in a row
ENGR00152309	[IPV_DMAv2] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00143256	[IPV_DSPI] - Code traceability must be checked against Design IDs	According to the process, the code traceability must be checked against design IDs, using "@remarks Implements DSPIxxxx" directives; only the situations when design IDs are not applicable must be traced against Requirement IDs At this moment, the SPI driver traceability is checked against Requirement IDs in almost all cases.
ENGR00139032	[IPV_DSPI] - MISRA violation on E_OK & E_NOT_OK	Attributions and comparisons between Std_ReturnType variables (uint8) and E_OK & E_NOT_OK values (signed integer) are violating MISRA, due to the type mismatch.

ENGR00143209	[IPV_DSPI] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00170007	[IPV_DSPI] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD 32 byte allignment nedds to be done for Cosmic compiler
ENGR00159693	[IPV_DSPI] Driver generates compiler warnings when compiled with DIAB 5.8.0.0	Problem detailed description (how to reproduce it): While compiling with DIAB 5.8.0.0, DSPI driver generates below compiler warnings. "narrowing or signed-to-unsigned type conversion found: " Preconditions: DIAB 5.8.0.0 compiler version shall be used for building driver. Trigger: [...] Observed behavior: Compiler warning is generated during building driver. When can it be observed? (at configuration time, at runtime, at compile time?) compile time 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 the compiler warning in driver files. If it can not be fixed, update the comments in driver code with proper doxygen tags.
ENGR00158597	[IPV_DSPI] Fix MISRA violation	NewWork Description: MISRA violation 8.8 need to be fixed for SPI driver. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the SPI driver files to fix MISRA violation 8.8.
ENGR00170099	[IPV_DSPI] Function prototypes to be mapped	Conformance tests results for the SPI drivers are attached. Please analyze them and fulfill the analysis report. 1. In Spi.c file local and global functions are not mapped. 2. Memory mapping is missing for Dspi_LLD.c and Spi.h file. 3. In Dspi_LLD.h file SPI_START_CONFIG_DATA_UNSPECIFIED is not closed properly.
ENGR00152300	[IPV_DSPI] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00152661	[IPV_DSPI] Porting to MCAL3.0, Bolero 256k	Using Bolero MCAL3.0 (512k,1.5M) v3.0.1 as a basis, the SPI module shall be ported to MCAL3.0 Bolero 256k in a new prodcut branch. Product name: Freescale AUTOSAR MPC560xD MCAL3.0 First release version: V0.9.0 (Beta) Tresos IDs: Arichtecture: 'PA', ID=2; derivative: ‘XPC560XD’, ID=23 Starting point for port: MCAL3.0, MPC560xB v3.0.1 release More details see SOW.
ENGR00162101	[IPV_DSPI] Spi_DataType shall be defined at generation time to save RAM usage	NewWork Description: Spi_DataType shall be defined at generation time to save RAM usage. this is a feature request coming from one of our customer: currently, the Spi_DataType is statically defined to uint16. However, as customer only uses 8bit buffers for SPI, they would waste RAM by allocating 16bit. Thus, they request to determine at generation time the maximum size of bits needed for that data type and use the smallest possible c data type. Expected behavior: Requirement source: Customer Request Proposed solution (Optional):
ENGR00141622	[IPV_DSPI] Timed Serial Bus (TSB) configuration to support Micro Second Bus	It is requested the implementation of a not-Autosar API to support the configuartion of the TSB in the "DSPI DSI Configuration Register (DSPI_DSICR)" to enable user to access the Micro Second Bus functionality. This feature is supported only by DSPI peripheral implemented in Power Train platforms (Monaco and Andorra). The chapter 25.5.9 of the attached reference manual describes the HW features of the TSB. Here below the SW sequence requested by the cvustomer to access the TSB functionality: In order to enable the MSC mode we need the following configurations: 1- DSICR1 register (DSPI_BASE + 0xD0) Full register configuration 2- MCR register: set field DCONF for CSI mode (not SPI) The other MCR configuration should be same as SPI: * master mode * Tx fifo enable * Rx fifo disable * continuous clock mode CONT_SCKE 3- CTAR[x] registers for frame definition & baudrate Then the SPI will be set and when started (Halt bit cleared in MCR register) the transmission of data is automatic and continuous After by SW some command frame must be sent For this it is enough just to write to the PUSHR register like this void SPI_B_CMD(vuint32_t CMD) { DSPI_B.PUSHR.R = CMD; }

ENGR00153649	[IPV_DSPI] Updates for Leopard RTM 1.0.1	NewWork Description: Update the files for MISRA violations for PCLint 9.00F. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140227	[IPV_EMIOS] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00141510	[IPV_EMIOS] Add a relevant comment for setting the CCR_EDSEL_MASK in eMIOS_LLD_Stops	In eMIOS_Icu_LLD.c, please comment REG_BIT_SET32(EMIOS_CCR(mIdx, chIdx), CCR_EDSEL_MASK); setting in eMIOS_LLD_StopSignalMeasurement
ENGR00154989	[IPV_EMIOS] Additional mapping of the logic channel names to the HW-channels	Context: In Pwm_Cfg.h, the logic channel names are mapped to channelIDs: #define PWM_CHANNEL0 0. In the Pwm_Cfg.c file, the HW-channels are used in the Pwm_ChannelConfigType structure: > EMIOS_0_CH_0 Problem: The mapping between logical and HW channels is calculated in PWM driver during runtime, which is inefficient; e.g. runtime of Pwm_SetDutycycle is approximately 2-4 µs. In comparison, a direct HW-register access, which would be possible with the above mentioned mapping, would take approximately 60ns. Suggestion: An additional mapping of the logic channel names to the HW-channels would help, i.e.: #define PWM_CHANNEL0_EMIOS_CH EMIOS_0_0 For improving the readability of the configuration files, this #define may be used in the Pwm_ChannelConfigType structure (Pwm_Cfg.c).
ENGR00140823	[IPV_EMIOS] FLAG bit in EMIOS_CSR is respoding in stop signal measurment	In stop signal measurment function, mode is changed to GPI mode, EDSEL is not set to 1, so the FLAG bit in EMIOS_CSR register is responding.
ENGR00155082	[IPV_EMIOS] General updates for Bolero RTM 3.0.2	NewWork Description: Update all the filesfor MISRA analysis: Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00141529	[IPV_EMIOS] Integrate the review comment updates	Integrate the updates after EMIOS reviews to mln line (new label BLN_IPV_EMIOS_MCAL_3.0_02.09.00 needed to be created due to previous Andorra release which used the previous IPV label).

ENGR00155122	[IPV_EMIOS] Period is one tick higher than requested	The register value of the period of counterbuses is one tick too high. If I choose for example 20kHz I get as output only 19,6kHz. This is because the PWM driver always adds 1 tick(line 782). (line768 Period = PwmChannel->Pwm_DefaultPeriodValue; //FIXME AT old code was + "+ 1U";) - For a more exact duty cycle there should be some kind of rounding in the function "EMIOS_PWM_LLD_ComputeDutycycle" in which the dutycycle is calculated. (line 450 //FIXME AT, old code dutycycle = (Period * Duty) >> 15; //FIXME AT, patch for exactness Calc_rounding = Period * Duty; if((Calc_rounding & 0x4000) == 0x4000) { dutycycle = (Calc_rounding>>15)+1; } else { dutycycle = Calc_rounding>>15; }) - For the duty cycle calculation the period value which is stored in "Pwm_EMIOS_LLD_Period" should nevertheless calculated with 1tick added.I100:I101 (line804 Pwm_EMIOS_LLD_Period[hw_ch] = Period+1; //FIXME AT, old code without +1)
ENGR00161221	[IPV_EMIOS] Request to manipulate trigger-delay on runtime	The PWM driver shall provide an optional function, API and configuration parameters to influence the trigger delay at runtime. This optional API shall be called "Pwm_SetTriggerDelay(uint16 TriggerDelay)" Per default these optional functions, APIs and configuration parameters shall be disabled.
ENGR00169368	[IPV_EMIOS]- comments or code should not exceed 100 characters	Problem detailed description (how to reproduce it): All the files for Icu driver needs to be check so that it will not exceed 100 characters [...] Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00160902	[IPV_EMIOS]-Update Misra violation for EMIOS (PWM)	NewWork Description: fixed misra violation for emiosr Fix misra violation for eMIOS_Pwm_LLD.c,eMIOS_Pwm_LLD.h […] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00163346	[IPV_EMIOS]-Update Misra violation for EMIOS (PWM)	NewWork Description: fixed misra violation for emios Fix misra violation for eMIOS_Pwm_LLD.c Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161499	[IPV_EMIOS]-Update Misra violation for IPV_EMIOS for ICU Files	NewWork Description: fixed misra violation for emiosr Fix misra violation for eMios_Icu_LLD.h […] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161837	[IPV_EMIOS]-Update driver files to implement the workaround for LDRA code coverage bug	NewWork Description: Update EMIOS files to update the comments before the "#endif" statement. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161494	[IPV_EMIOS]: Correct variable should be check to clear the EMIOSB UC register	Problem detailed description (how to reproduce it): Variable "reg_EMIOSB_available" which indicates the availability of register EMIOS UC B should be used instead of variable "reg_EMIOSCNT_available" to Clear the EMIOSB UC register if exists. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00151669	[IPV_EMIOS]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]

ENGR00162932	[IPV_EMIOS]: The new Non-Autosar API Pwm_SetTriggerDelay() is not protected under the gua	Problem detailed description (how to reproduce it): The new Non-Autosar API Pwm_SetTriggerDelay() is applicable only for OPWMT mode so it must be put under the guard for this mode. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00139887	[IPV_EQADC] Adc_ReadGroup and Adc_GetStreamLastPointer does not reset the ADC_Groups	Following the State Diagram of Streaming Access Mode, when both Adc_ReadGroup and Adc_GetStreamLastPointer API are called while the conversion is ADC_COMPLETED, the state must return in ADC_BUSY as done but the GroupStatus[Group].ResultIndex must be reset to 0.
ENGR00170317	[IPV_EQADC] Adc_ReadGroup optimization	NewWork Description: Adc_ReadGroup optimization Expected behavior: Based on actual implementation a typical reda sequence is going trough the following steps: Adc_Init(); Adc_SetupResultBuffer(GroupNumber, &ResultBuffer); Adc_StartGroupConversion(GroupNumber); While(ADC_State < BUSY){} Adc_ReadGroup(GroupNumber, &UserBuffer); ….. An optimized version of the Adc_ReaGroup could avoid to loop on all the channel inside the group and copying the result directly inside the expected result buffer. Then the last step in the previous sequence can be cahnged passing directly the user destination buffer Adc_ReadGroup(GroupNumber, &ResultBuffer); This optimization can be applied only if the stream depth is limited to 1. Requirement source: Customer Request
ENGR00151620	[IPV_EQADC] Calibration constants updated only in ADC standard registers	The issue is that if calibration is enabled, the calculated calibration constants are written only to standard configuration registers (ADC reg address 4 and 5). Please see code line 637 in EQADC_LLD.c. /* write command messages to write cal_gcc and occ calibration constants into */ REG_WRITE32((EQADC_CFPR0(eQADC_AorB)+((uint32)(fifoindex)<<2U)), (c_buffer | (((uint32)cal_gcc) << 8UL) | ADC_GCCR)); REG_WRITE32((EQADC_CFPR0(eQADC_AorB)+((uint32)(fifoindex)<<2U)), (c_buffer | ((((uint32)occ) & ADC_OCC_REG) << 8UL) | ADC_CAL_CONST | ADC_OCCR)); If the 10 bits or 8bits conversion is used the calibration does not take effect. Therefore the constants should be written also to the alternate1 and alternate2 register sets as well (ADC reg address 31,32 and 35,36).
ENGR00170183	[IPV_EQADC] Cosmetic Adc driver improvements	Several cosmetic Adc driver improvements, for all platforms, can be done. Please see the analysis tab.

ENGR00151543	[IPV_EQADC] DMA initialization for QADC	========== File: EQADC_LLD.c @ plugins\Adc_TS_T2D14M1I9R0\src Function name: EQADC_StartDMAOperation: Line2103 of file { …… DestOffset = (uint32)(sizeof(Adc_ValueGroupType) * (uint32)GroupPtr->NumSamples); …… DlastDataValue = (uint32)((2U * (uint32)((uint32)((uint32)(GroupPtr->NumSamples) - 1U) + ((uint32)((uint32)ChannelsInGrp - 2U) * (uint32)(GroupPtr->NumSamples)))) + (uint32)DestOffset); } Condition: NumSamples = 1, ChannelsInGrp = 16 Result with theory: DlastDataValue = 30, but from Monaco DMA side, the value should be 32 Practical result: the value of the TCD.DADDR was changed, and the sampled value was shift into the buffer with different beginning address each time.
ENGR00170176	[IPV_EQADC] Incorrect memory mapping	Problem detailed description (how to reproduce it): Please check if all drivers' symbols are placed in between START/STOP sections present on MemMap.h file. Usually these sections are the following: drv_START_CONFIG_DATA_UNSPECIFIED drv_STOP_CONFIG_DATA_UNSPECIFIED drv_START_SEC_CODE drv_STOP_SEC_CODE drv_START_SEC_VAR_UNSPECIFIED drv_STOP_SEC_VAR_UNSPECIFIED drv_START_SEC_CONST_UNSPECIFIED drv_STOP_SEC_CONST_UNSPECIFIED The following findings were raised by our clients. Even if the current driver is not recorded in the list below please check the consistency. > The following symbols are missing a correct memory mapping. All of these symbols get assigned to section ".text_vle": > MCAL (Leopard BETA 0.9.1, but also RTM1.0.0) > * Functions: > * CAN_BASE_ADDRS: Not found at all > * DIO_SIUL_ReadChannel(), > DIO_SIUL_WriteChannel(), > DIO_SIUL_ReadPort(), > DIO_SIUL_WritePort(), > DIO_SIUL_ReadChannelGroup(), > DIO_SIUL_WriteChannelGroup(): > + Dio_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Dio_TS_T2D17M0I9R0\include\Siul_LLD_dio.c: Not mapped > + Port_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Port_TS_T2D17M0I9R0\src\Siul_LLD_port.c: Not mapped
ENGR00152927	[IPV_EQADC] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00143448	[IPV_EQADC] Typecast error when Dma_configure_channel is used.	Problem detailed description (how to reproduce it): I just checked the IPVault EQADC_LLD.c file clear case version-16 and it seems that this is due to the change of typecast from previous version- 15(uint8 to DmaTcdPointerType)which is highlighted below was done on latest version of the file. "D:/EB_Fado/plugins/Adc_TS_T2D19M1I0R0/src/EQADC_LLD.c", line 2179: error #31: expression must have integral type Dma_configure_channel((DmaTcdPointerType)((fifoindex*2U) + (eQADC_AorB*EQADCB_DMACH_START)), /* dma_channel */ ^ "D:/EB_Fado/plugins/Adc_TS_T2D19M1I0R0/src/EQADC_LLD.c", line 2195: error #31: expression must have integral type Dma_configure_channel((DmaTcdPointerType)((fifoindex*2U) + (eQADC_AorB*EQADCB_DMACH_START) + 1U), /* dma_channel */ Preconditions: . Trigger: . Observed behavior: Compilation error. Expected behavior: No compilation error. 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 typecast from DmaTcdPointerType to unit8. Re-put the version 15 instead of 16.
ENGR00169566	[IPV_ESCI] - Request to add LIN_ prefix to macro used in ESCI_LLD.c file	Modification request on the ESCI_LLD.c file: Few macros used in ESCI_LLD.c file needs to be modified to add LIN_ prefix which are defined in Lin.h file and got modified in ENGR00155121. In ENGR00155121, LIN_ prefix is added to the macros mentioned below in Lin.h header file /** @brief Interrupt Errors conditions */ #define NO_ERROR 0x00U #define BIT_ERROR 0x01U #define CHECKSUM_ERROR 0x02U #define SYNCH_FIELD_ERROR 0x03U #define BREAK_DELIMITER_ERROR 0x04U #define IDENTIFIER_PARITY_ERROR 0x05U #define FRAMING_ERROR 0x06U #define BUFFER_OVER_RUN_ERROR 0x07U #define NOISE_ERROR 0x08U /* Commands IDs */ #define TX_MASTER_RES_COMMAND 0x01U #define TX_SLAVE_RES_COMMAND 0x02U #define TX_SLEEP_COMMAND 0x03U #define TX_NO_COMMAND 0x04U
ENGR00152789	[IPV_ESCI] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver

ENGR00139913	[IPV_ESCI] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00141443	[IPV_ESCI] Configuration enhancement to support Micro Second Bus upstream channel	The LIN implementation on the eSCI peripheral shall support the capability to configure the serial channel as upstream channel for th Micro Second Bus. The eSCI is used as receiver and the supported frames format shal be either the 16-bit or the 14bit MSC upstream frames.
ENGR00152520	[IPV_ESCI] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.
ENGR00138815	[IPV_ESCI] Hardware channel offset calculation for eSCI_J to eSCI_M lin channels is wrong	The register(s) calculation for eSCI_J to eSCI_M lin channels is not correct in Reg_eSys_ESCI.h file. The base address of eSCI_J is 0xC3FA0000 which is not consecutive with offset (0x4000) from eSCI_A base address i.e 0xFFFA0000. File: Reg_eSys_ESCI.h
ENGR00138814	[IPV_ESCI] Missing "Lin_BuffType" structure for eSCI_D to eSCI_M lin channels in variable "Lin	The Lin_BuffType structure is implemented for only eSCI_A, eSCI_B and eSCI_C lin channels, but not for remaining lin channels i.e eSCI_D to eSCI_M. File: ESCI_LLD.c
ENGR00152931	[IPV_ESCI] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00161639	[IPV_ESCI] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00138310	[IPV_ESCI] Remove compilation warning message of LIN driver (ESCI_LLD.c)	Variable "Lin_ESCI" was set but not used in the function ESCI_LLD_HardwareGetStatus(), so this variable can be removed from the function. Reported Component Baseline: BLN_IPV_ESCI_MCAL_3.0_01.02.00 file: ESCI_LLD.c The issue exists also in the latest IPVault component label (BLN_IPV_ESCI_MCAL_3.0_01.04.00).
ENGR00143961	[IPV_ESCI] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).

ENGR00162336	[IPV_FLASH] Fls DSI specific defines shall be renamed in IPV_FLASH.	NewWork Description: The following defines and types shall be defined by FSL driver: /*** **** DEFINES ***/ #define EXC_UNHANDLED 0 #define EXC_HANDLED_RETRY 1u #define EXC_HANDLED_SKIP 2u #define EXC_HANDLED_STOP 3u /*** **** TYPEDEFS ***/ typedef uint8_least Exc_CompHandlerReturnType; typedef const uint8 * Exc_InstructionAddressType; typedef const void * Exc_DataAddressType; typedef struct { Exc_InstructionAddressType instruction_pt; Exc_DataAddressType data_pt; uint32 syndrome_u32; } Exc_ExceptionDetailsType; typedef const Exc_ExceptionDetailsType *ExceptionDetailsPtrType; /***/ Expected behavior: […]
ENGR00140292	[IPV_FLASH] - Added define for ANDORRA 2M for MCU driver	Should be added define for ANDORRA 2M in order to configure flash general register in Mcu driver.
ENGR00140738	[IPV_FLASH] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).

ENGR00143252	[IPV_FLASH] Add FLASH KOMODO support for Mcu driver	NewWork Description: Should be updated the functionality to initialize the FLASH settings for KOMODO: - Address Pipelining Control, - Write Wait State Control, - Read Wait State Control, - Read-While-Write Control, - Page Buffer Configuration, - Page Buffer Configuration, - Data Prefetch Enable, - Instruction Prefetch Enable, - Prefetch Limit, - Buffer Enable, - Arbitration Mode, - Master Prefetch Disable - Master Access Protection. Expected behavior: Should be updated Mcu_Flash_Init and Mcu_Flash_Configure functions. Requirement source:MCU116 - AUTOSAR Specification of MCU driver (AUTOSAR_SWS_MCU_Driver.pdf)
ENGR00141699	[IPV_FLASH] Add FLASH PICTUS 1M support for Mcu driver	Should be updated the functionality to initialize the FLASH settings for PICTUS 1M: - Address Pipelining Control, - Write Wait State Control, - Read Wait State Control, - Read-While-Write Control, - Page Buffer Configuration, - Page Buffer Configuration, - Data Prefetch Enable, - Instruction Prefetch Enable, - Prefetch Limit, - Buffer Enable, - Arbitration Mode, - Master Prefetch Disable - Master Access Protection.
ENGR00144131	[IPV_FLASH] Add support for bolero 256k	NewWork Description: Add support for bolero 256k Expected behavior: Support for Bolero256 Proposed solution (Optional): Update following files: Flash_Mcu_LLD.c: added support for BOLERO256 platform Reg_eSys_FLASHC.h: added support for BOLERO256 platform Reg_FlashMem.h added following define #define IPV_FLASH_C90FG_256_1_4_D 0x02561444UL /* for XPC5602D platform */
ENGR00142007	[IPV_FLASH] Added Komodo IPV_FLASH_C90FG_1024_1_8 macro define	Added Komodo IPV_FLASH_C90FG_1024_1_8 macro define in Reg_FlashMem.h
ENGR00139961	[IPV_FLASH] Added macro define for ANDORRA 2M	Added macro define for ANDORRA 2M in the Reg_FlashMem.h file
ENGR00141501	[IPV_FLASH] Added macro define for PICTUS 1M	Added macro define for Pictus 1M into Reg_FlashMem.h

ENGR00153119	[IPV_FLASH] Compiler warnings to be corrected for Leopard	Problem detailed description (how to reproduce it): We received a new report dealing with some compiler warnings that shall be removed. The type cast in some function calls (Dem_ReportErrorStatus) shall be changed: from "uint16" to "Dem_EventIdType" in the following files: FlexCan_LLD.c Fls.c Modes_LLD.c Monitor_LLD.c We have " FAULT_MCU_E_TIMEOUT_TRANSITION" defined as a "enum" and with the casting you currently have we get warnings: "MCAL\41516132_Modes_LLD.c", line 255: warning (etoa:4188): enumerated type mixed with another type Dem_ReportErrorStatus((uint16) FAULT_MCU_E_TIMEOUT_TRANSITION, DEM_EVENT_STATUS_FAILED); Please make sure that other modules are checked as well. Preconditions: NA Trigger: NA Observed behavior: Compiler warnings When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: No compiler warnings for MCAL code. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The type cast in some function calls (Dem_ReportErrorStatus) shall be changed:
ENGR00156139	[IPV_FLASH] Development for Leopard 2M (EAR0.8.0)	NewWork Description: The implementation of the Mcu driver for the platform SPC56EL70 requires the IPV_FLASH to support the correct flash module version . Expected behavior: IPV_FLASH should support the Flash module version of the SPC56EL70 platform Requirement source: See SPC56EL60 Reference Manual Rev. 7, June 2011 Proposed solution (Optional): Changes affect the file Reg_eSys_FLASHC.h updated with the define that specifies the flash module version : #define IPV_FLASH_C90FL_2048_4_0_Nh 0x2048406EUL
ENGR00143560	[IPV_FLASH] ECC handling when interrupts are disabled	Problem detailed description (how to reproduce it): For Bolero (z0 core) the FLS driver provides the DSI handler for ECC support. This mapps to IVOR2. However, if the interrupts are disabled (MSR[EE] = 0), IVOR1 is generated instead. This is not supported in the FLS handling. Preconditions: MSR[EE] =0 Trigger: ECC ->IVOR1 Observed behavior: ECC is not recognized by the FLS driver. Expected behavior: ECC must be recognized regardless of the interrupts state. Proposed solution (Optional): N/A
ENGR00144269	[IPV_FLASH] Fls_Ac.c does not check SW version	Fls_Ac.c does not check SW version for the included headers Issue found on AUTOSAR MPC56XXA MCAL3.0 swv: 2.0.0 RTM

ENGR00143543	[IPV_FLASH] Fls_Cancel does not abort interleaved block sector	Problem detailed description (how to reproduce it): Using a configuration with: - An interleved block sector - Call Fls_Cancel for erase-abort operation on interleaved block (asynch mode) The Fls_Cancel does not manage the abort setting for interleaved selector register Preconditions: Interleaved block is configured for asynch erase Trigger: [...] Observed behavior: [...] Expected behavior: Fls_Cancel on interleved block works properly Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Added code to manage the erase-abort operration on interleved block in the abort condition
ENGR00151782	[IPV_FLASH] Fls_Cancel should abort all kind of flash sectors	NewWork Description: Using a configuration with: - For all block sectors - Call Fls_Cancel for erase-abort operation on interleaved block (asynch mode) The Fls_Cancel should cancel an ongoing job Expected behavior: Fls_Cancel works properly on all kind of flash sectors Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00133618	[IPV_FLASH] Fls_Erase (async) fails for Flash_A and Flash_B (High-address space) sectors	During the integration of the AutoCore for customer on XPC564XA we found a bug in the Erase routine in the current release of the Fls driver. Our project requires the configuration of Fls sectors: FLS_ARRAY_0_A_1_B_PART_07_H00, FLS_ARRAY_0_A_1_B_PART_08_H01, FLS_ARRAY_0_A_1_B_PART_09_H02, FLS_ARRAY_0_A_1_B_PART_10_H03, FLS_ARRAY_0_A_1_B_PART_11_H04, FLS_ARRAY_0_A_1_B_PART_12_H05 All sectors are in Flash_A and Flash_B (High-address space). Because of timing issues we expect the value "FlsSectorEraseAsynch" set to "true". Now when trying to erase these sectors this will fail. The sector will not be erased completely, please see the Trace32 debugger screen shot after performed erase action (see the attached picture). So we checked function "Fls_LLD_SectorErase" and found out that in case of synchronous mode is used: 1) erase instructions for Flash_A are performed 2) erase instructions for Flash_B are performed asynchronous mode is used: 1) erase instructions for Flash_A are performed In asynchronous mode there is no instruction for setting the flags for Flash_B. In function "Fls_LLD_MainFunction" in case of asynchronous erase, only Flash_A erase checks are performed. 1) Please check erase-functionality of the used Fls driver for Flash_A and Flash_B (High-address space). Give some feedback according the described behavior. 2) In case that you confirm this is an error, please give feedback to possible bug-fix schedule.
ENGR00142285	[IPV_FLASH] Linear address not well managed in case of Async write for FlsMaxWriteNormalMo	Problem detailed description (how to reproduce it): Using a configuration with: - Flash blocks not contiguous - FlsMaxWriteNormalMode = 32 - Erase and Write job in ASYNC mode writing 32 across two non contiguous block a writing error is detected Preconditions: [...] Trigger: [...] Observed behavior: Only 8 byte are correctly written by Fls_MainFunction() Expected behavior: 32 byte shopuld be correctly written by Fls_MainFunction() Note: in the “Expected behavior” field, please mention also the requirement source too Proposed solution (Optional): Only in case of Async Write job update Fls_JobSectorIt at the end of Fls_LLD_MainFunction.
ENGR00138218	[IPV_FLASH] Macro definition "RESET_PFCR_APC_WWSC_RWSC" is missing	Macro definition "RESET_PFCR_APC_WWSC_RWSC" is removed from Reg_eSys_FLASHC.h, while fixing ENGR00132386, which is being used by "Mcu_LLD.c" file and causing compilation error Reported Component Baseline: BLN_IPV_FLASH_MCAL_3.0_01.16.01 file: Reg_eSys_FLASH.h

ENGR00152939	[IPV_FLASH] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00152966	[IPV_FLASH] Pictus 1M support not considred in the file Reg_eSys_FLASHC.h	The support for Pictus 1M in file Reg_eSys_FLASHC.h is missing. On line 113 is #ifdef IPV_FLASH is only for Pictus 256/512 and not for Pictus1M. This MCAL is intended to support Pictus1M. See attached screenshot with mine fix for this issue.
ENGR00158203	[IPV_FLASH] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]

ENGR00162471	[IPV_FLASH] Provide the source code for the Fls_ac.c file	NewWork Description: There is a customer request to provide source code for Fls_ac.c file. Would it be possible to include this source code directly in the MCAL distribution (it can be provided in a comment section within the c file) This is usually needed for debugging purpose... To wait after ENGR00162640 is completed Expected behavior: NA Requirement source: Customer Request Proposed solution (Optional): Would it be possible to include this source code directly in the MCAL distribution (it can be provided in a comment section within the c file)
ENGR00151870	[IPV_FLASH] System is locked during Fls_LLD_SectorErase (WDG callback implementation)	“What we need for integration is a Fls driver that erase function works either in async mode or sync mode, but with triggering the Watchdog. To solve this issue we already suggested to add a WDG callback function that will be triggered within erase function. This was done in previous versions of the MCAL for Pictus and Bolero. Otherwise we need the a async function for erase.” To note that the mentioned MCAL versions are the 2.1 Bolero and the ST codebase for Pictus. Based on this latest info, I assume you already know what ST has implemented in terms of FLS-WDG for the Pictus ST codebase. Hello, during the integration of the FLS we run in problems when erasing the FLS sectors. After requesting an erase of the sector we will run in a WDG timeout. We need to use the FLs in async mode., so we would expect that there is no lock . The WDG is configured to 20 ms. After debugging the function Fls_LLD_SectorErase we found out that following sequnce within this function takes more than 200 ms: /* back-up PFCR0 or PFCR1 */ Fls_LLD_pfcConfRegValue = Fls_LLD_pfcRegBasePtr[Fls_LLD_pfcConfRegOffset]; /* start internal erase/program sequence */ Fls_LLD_regBasePtr[FLASHMEM_MCR] |= MCR_EHV; Fls_LLD_Job = FLS_LLD_JOB_ERASE; Fls_LLD_JobResult = MEMIF_JOB_PENDING; During this time the WDG is not triggered, the system performs a reset. Attached you can find the configurations used. Do we need to configure something else or why does the system take so much time? Please check this issue and give some feedback.
ENGR00138580	[IPV_FLASH] The parameter "physicalSectorSize" in "Fls_LLD_SectorErase( )" function needs to	The parameter "physicalSectorSize" will only be used by "Fls_LLD_SectorErase( )" function if the precondition "FLS_ERASE_BLANK_CHECK" is turned ON Reported component baseline: BLN_IPV_FLASH_MCAL_3.0_01.09.00 Origin tag: HF_MCAL_3.0_BOLERO_RTM_HF4_3.0.0 File: Fls.c
ENGR00141693	[IPV_FLASH] Unlock Middle address space for interleave block	Unlock Middle address space for interleave block

ENGR00142449	[IPV_FLASH] Unlock Middle address space should work for all kind of flash	NewWork Description: Unlock Functionality should work also for Middle address space for all kind of flash Expected behavior: Flash Unlock should work for standard flash and for Interleaved one Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140146	[IPV_FLASH] Unlock functionality does not work with interleaved blocks on High Address space	Unlock functionality does not work with interleaved blocks
ENGR00142447	[IPV_FLASH] Unlock functionality should works for all kind of Flash on High address Space	NewWork Description: Unlock functionality does not works for all kind of Flash (interleaved block for example) Expected behavior: flash sector are unlocked for all sectors in Low Middle and High address space for Normal flash and Interleaved one. Requirement source: PR-MCAL-3158 Proposed solution (Optional): [...]
ENGR00151785	[IPV_FLASH] Unusefull clear of select register during a sync erase for all kinf of blocks	NewWork Description: during a sync erase select register is cleared two times Expected behavior: Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): remove one clear from code.
ENGR00143553	[IPV_FLASH] Unusefull clear of select register during a sync erase on interleaved blocks	Problem detailed description (how to reproduce it): In the erase operation, on the interleaved block, the select register is clean twice Preconditions: Interleaved block is configured for asynch erase 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): Removed unsefull line
ENGR00142450	[IPV_FLASH] Write job do not work when FlsMaxWriteNormalMode>8 for all kind of flash	NewWork Description: Write operation in asynch mode should work when FlsMaxWriteNormalMode>8 on all kind of flash Expected behavior: One Fls_MainFunction should write FlsMaxWriteNormalMode byte for all kind of flash (Standard and interleaved) when FlsMaxWriteNormalMode > 8 Requirement source: Autosar Requirement Proposed solution (Optional): [...]

ENGR00142286	[IPV_FLASH] Write job do not work when FlsMaxWriteNormalMode>8 on Interleaved blocks	Problem detailed description (how to reproduce it): Write operation in asynch does not work when FlsMaxWriteNormalMode>8 on Interleaved blocks Preconditions: FlsMaxWriteNormalMode =32 Write Asynch = TRUE Configured interleaves block Trigger: [...] Observed behavior: Writing 8 bytes and fail 24 Expected behavior: Write 32 bytes in the Fls_LLD_MainFunction Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00141366	[IPV_FLASH] Wrong define name during access pipeling	What happens (symptoms): ---------- Mcu_Flash_Configure_Ram function use wrong define name: FLASH_BIUCR0 instead of CFLASH_PFCR0 only for Pictus platform. When does this happen: ---------- Compilation error is generated on Flash_Mcu_LLD.c file line 315.
ENGR00152724	[IPV_FLASH] Wrong parameter for Dem_ReportErrorStatus in Fls_LLD_SectorWrite function (in	Problem detailed description (how to reproduce it): Not correct parameters are used for Dem_ReportErrorStatus function when the following condition are verified in the Fls_LLD_SectorWrite function: - Verify if EHV could be set for interleaved block - Verify if the sector is not protected against erase/program operations Preconditions: interleaved block Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): Used the FLS_E_WRITE_FAILED parameter (instead FLS_E_ERASE_FAILED) in the Dem_ReportErrorStatus when the these conditions are verified for interleaved block: if EHV cannot be set and sector is not protected against erase/program operations

ENGR00157415	[IPV_FLASH] compiler warning when Fls_Cancel api is disabled	Problem detailed description (how to reproduce it): following warning is displayed when copmpiling with FlsCancelApi=false: warning #177-D: function "Fls_LLD_Cancel" was declared but never referenced STATIC FUNC(void, FLS_CODE) Fls_LLD_Cancel(void) Preconditions: set FlsCancelApi=false in Fls.epc Trigger: [...] Observed behavior: A warneing is displayed When can it be observed? (at configuration time, at runtime, at compile time?) at compile time Expected behavior: only CER warning should be dispalied Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]
ENGR00160438	[IPV_FLASH] update Spectrum Flash IP version	NewWork Description: Spectrum IP version to be updated for porting FLS driver from Bolero RTM 3.0.2
ENGR00162640	[IPV_FLASH] while loops without timeout and DEM reporting in Fls_LLD_Cancel	1) In the below mentioned while loop in Fls_LLD_Cancel, the timeout is not present. while(!(Fls_LLD_regBasePtr[FLASHMEM_MCR] & MCR_DONE)) { /* wait until end of abort * } 2) Also the DEM error should be present in case of timeout. Both points are valid for all three while loops in Fls_LLD_Cancel function.
ENGR00151463	[IPV_FLASH]: MISRA Violation comments are missing for some of the files	NewWork Description: [MISRA Violation comments are missing for some of the files] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00151670	[IPV_FLASH]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00169613	[IPV_FLASH]Support Monaco platform in the file Fls_LLD_Code.m4	Support Monaco platform(IPV_FLASH_C90FL_640_34_61) in the file Fls_LLD_Code.m4

ENGR00144310	[IPV_FLEXCAN] - Implement the usage of RXGMASK, RX14MASK and RX15MASK	NewWork Description: [The registers RXGMASK, RX14MASK and RX15MASK are provided for legacy support and for low cost MCUs that do not have the individual masking per Message Buffer feature.] Expected behavior: [RXGMASK is used as acceptance mask for all Rx MBs, excluding MBs 14–15, which have individual mask registers RX14MASK and RX15MASK respectively. When the FEN bit in MCR is set (FIFO enabled), the RXGMASK also applies to all elements of the ID filter table, except elements 6-7, which have individual masks RX14MASK and RX15MASK respectively.] Requirement source: [ASR SWS] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
ENGR00160633	[IPV_FLEXCAN] Bolero3M errata review	NewWork Description: [review Bolero 3M errata dependent on silicon used for release 1. (cut 1.0) MPC5646BC_0M87Y Errata (19 AUG 2011) 2. (cut 2.0) MPC5646BC 0N32E Errata (19 AUG 2011)] Expected behavior: [create other CR in case of needed changes due to errata and used silicon for release] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140044	[IPV_FLEXCAN] CAN Driver HOH Numbering	4.1.2.3 CAN Driver HOH Numbering The CAN Driver configuration must support the HOH numbering as decided in Bugzilla issue #11714: The numbering for HTHs and HRHs together shall be unique and sequential within the driver and start from 0. Restriction: The HRHs and the HTHs shall be grouped, where HRHs come first. Example: HRH0 - 0, HRH1 - 1, HTH0 - 2, HTH1 - 3 This must especially be supported for configurations containing several controllers what in this case means that the HRHs of all controllers are grouped before the HTHs of all controllers (there must not be any HTH of any controller with an object ID lower than that of any HRH configured for this driver). Please also refer to this Autosar bugzilla entry: http://www.autosar.org/bugzilla/show_bug.cgi?id=11714#c23
ENGR00142155	[IPV_FLEXCAN] CAN driver stays more then 6us in the demask loop	within the first running benchmark of Bolero 3M COM Stack we discover an interesting timing behavior in the CAN driver during incoming CAN RX message 1) The CAN driver try to demask the right message from the message buffer for signals (32-63) behind single ISR Routine shall service further all flags from message buffer 2) COM Stack start up and raise the identified signal (CANIF, IPUm) 3) After finishing the signal raise the CAN driver stay more then 6us in the same demask loop (more run time than summerize CANIF, IPUm and COM needs) This behavior is not understandable for us. It seams the driver checks all signals up to number 63. Questions: Can somebody on FSL side explain to us what is the background of that? I believe we are simply scrolling thru all Rx messge buffers from 32 to 63 but could you please confirm or get me more details?

ENGR00160346	[IPV_FLEXCAN] CAN second controller not functional when the multiplex transmission is on	Problem detailed description (how to reproduce it): The CAN driver behavior for multiplex transmissions was changed during other changes done for ENGR00144127. By this change, the second CAN controller seems to no longer work. Preconditions: Multiplex transmission on first controller Trigger: NA Observed behavior: After updating the IDs per the clarifications of the ENGR00144127, the multiplexed transmission works fine now, however the second CAN controller is no longer transmitting any message. The customer tried at run time with a debugger to modify the object Ids on the second controller back to what they were with the “non working config” and in that case transmission is Ok Could you check if such a scenario has been tested? By updating by hand the IDs to be in the space of the second controller, it seems the second cotroller starts transmitting. (so it might be an issue with the ID >31 being assumed as part of controller 2 and <=31 as controller 1. Since by mulitplex transmission several MBs share the same ID, that some MB with ID <31 will be part of controller 2 which seems to break the driver. Runtime Expected behavior: Second controller working Proposed solution (Optional): NA
ENGR00143874	[IPV_FLEXCAN] Can internal flag not reset before CanIf_TxConfirmation	Problem detailed description (how to reproduce it): In a CanIf_TxConfirmation, a new Can_Write request can be done syncrhonously. So, the Busy flag and IT flag cmust be cleared before call Tx confirmation else the can_write will be rejected as Busy. Preconditions: NA Trigger: NA Observed behavior: NA Expected behavior: NA Proposed solution (Optional): In Can_LLD_Process_Tx: add these line befor the switch ((uint32)FLEXCAN_MB_READ_CS(can_hw_offset, can_mb_index)) //debug AG #if (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_64) FLEXCAN_IFLAG_CONFIG(can_hw_offset, can_iflag_reg_index, can_temp_flag); #else /* (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_32) */ FLEXCAN_IFLAG1_CONFIG(can_hw_offset, can_temp_flag); #endif /* (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_32) */ and add these lines befor call to CanIf_TxConfirmation: #if (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_64) Can_ControllerStatuses[controller].TxGuard[can_iflag_reg_index] &= (uint32)(~can_temp_flag); #else /* (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_32) */ Can_ControllerStatuses[controller].TxGuard[0] &= (uint32)(~can_temp_flag); #endif /* (CAN_MAXMB_SUPPORTED == FLEXCAN_MB_32) */

ENGR00152610	[IPV_FLEXCAN] Compiler warnings to be corrected for Leopard	Problem detailed description (how to reproduce it): We received a new report dealing with some compiler warnings that shall be removed. The type cast in some function calls (Dem_ReportErrorStatus) shall be changed: from "uint16" to "Dem_EventIdType" in the following files: FlexCan_LLD.c Fls.c Modes_LLD.c Monitor_LLD.c We have " FAULT_MCU_E_TIMEOUT_TRANSITION" defined as a "enum" and with the casting you currently have we get warnings: "MCAL\41516132_Modes_LLD.c", line 255: warning (etoa:4188): enumerated type mixed with another type Dem_ReportErrorStatus((uint16) FAULT_MCU_E_TIMEOUT_TRANSITION, DEM_EVENT_STATUS_FAILED); Please make sure that other modules are checked as well. Preconditions: NA Trigger: NA Observed behavior: Compiler warnings When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: No compiler warnings for MCAL code. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The type cast in some function calls (Dem_ReportErrorStatus) shall be changed:
ENGR00159691	[IPV_FLEXCAN] Driver generates compiler warnings when compiled with DIAB 5.8.0.0	Problem detailed description (how to reproduce it): While compiling with DIAB 5.8.0.0, FLEXCAN driver generates below compiler warnings. "narrowing or signed-to-unsigned type conversion found: " Preconditions: DIAB 5.8.0.0 compiler version shall be used for building driver. Trigger: [...] Observed behavior: Compiler warning is generated during building driver. When can it be observed? (at configuration time, at runtime, at compile time?) compile time 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 the compiler warning in driver files. If it can not be fixed, update the comments in driver code with proper doxygen tags.
ENGR00143069	[IPV_FLEXCAN] FlexCan_LLD.c uses a macro from Cer.h but does not include this file.	FlexCan_LLD.c uses a macro from Cer.h but does not include this file. The label used to release the module in the SPC560xP sample drop (named EAR 2.8.0) is: INT_CAN_MCAL_3.0_XPC560XP_01.01.00_I01.

ENGR00153425	[IPV_FLEXCAN] HOH check for Tx/Rx order is done on list index and not on Can Object IDs	Problem detailed description (how to reproduce it): On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Error: [...\tresos\plugins\Can_TS_T2D13M3I0R0/generate_PC/include/Can_Cfg.h:351]: The HRH and HTH Ids are defined under two different name-spaces (Refer to CAN326). Example: HRH0-0, HRH1-1, HTH0-2, HTH1-3. The HRHs of all controllers are grouped before the HTHs of all controllers (there must not be any HTH of any controller with an object ID lower than that of any HRH configured for this driver). Note that the object itself does not change, only the position in the list changes. Preconditions: NA Trigger: Configuration time error. Observed behavior: On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Expected behavior: CAN326 refers to IDs, not indexes Proposed solution (Optional): see solution implemented in ENGR00131632
ENGR00142933	[IPV_FLEXCAN] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.
ENGR00153386	[IPV_FLEXCAN] Identify implemented workarounds for each platform	We need to identify implemented workarounds for each driver/platform and fulfilling attached file Implemented_Workaround_Review.xlsx. What could help with identifying : 1.User Manual documentation checking 2. Checking of previous errata analysis and CR's used for workarounds implementation (list in attached file) 3. Checking whether other developers are aware of any "special" code implementation
ENGR00139096	[IPV_FLEXCAN] MB are locked when using Canif_TxConfirmation	When CanIf_TxConfirmation is used as trigger for further can messages (eg TP) the current can driver implementation lead to a blocked MB. The relevant Register flags were not cleared before calling CanIf_TxConfirmation (FlexCan_LLD.c Line 2321) The confirmation is called without clearing IFLAG1 and the TxGuard. Note: Also for CancleTxConfirmation the Flags need to be reseted
ENGR00163728	[IPV_FLEXCAN] Memory mapping	Conformance tests results for the CAN drivers are attached. Please analyze them and fulfill the analysis report. Memory mapping is done in FlexCan_LLD.c and Can_LLD.c, it should be applied to FlexCan_LLD.h and Can_LLD.h.
ENGR00151768	[IPV_FLEXCAN] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).

ENGR00156761	[IPV_FLEXCAN] Review Pictus 256k/512k erratas	NewWork Description: Analyze - Pictus512K Errata Report for Cut3.4 (see it attached) and - Pictus256K Cut1.1 Errata Report (see it attached) check impact on MCAL drivers/ IPVs. Expected behavior: HW failure shall be invisible to the customer. Requirement source: PICTUS512K_1M36W errata sheet from 07.07.2011 PICTUS256K_0M22Y_2011-08-19_PDMonly_Preliminary.pdf from 19.08.2011 Proposed solution (Optional): Implement required software workarounds
ENGR00153639	[IPV_FLEXCAN] Update the MISRA violations for PCLint 9.00F	NewWork Description: Update the MISRA violations for PCLint 9.00F Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00152874	[IPV_FLEXCAN] Update the stubs, to no longer export any BASE headers	NewWork Description: - Update the stubs, to no longer export any BASE headers Expected behavior: - No BASE headers are visible by including the stubs headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00152920	[IPV_FLEXCAN] Wrong addressing on structure MessageBufferConfigsPtr	Problem detailed description (how to reproduce it): The structure can_ptrMBConfigContainer->MessageBufferConfigsPtr is wrong addressed when it wants to write in a register with FLEXCAN_RXIMR_CONFIG macro, in function Can_LLD_InitController.(ex: line 864, 919). Preconditions: NA Trigger: NA Observed behavior: The structure can_ptrMBConfigContainer->MessageBufferConfigsPtr is addressed with an index that is out of bounds(can_mb_ctrl_index). This structure keeps the MBs that are given by the user. But when the RxFifo is enabled the index can_mb_ctrl_index is incremented with 8. So the addressing of the structure is wrong. CAN_BCC_SUPPORT_ENABLE functionality is also impacted. When can it be observed? (at configuration time, at runtime, at compile time?) At runtime Expected behavior: The structure can_ptrMBConfigContainer->MessageBufferConfigsPtr should be addressed with the correct index (can_mb_global_index). Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The addressing of the structure should be done with the correct index(can_mb_global_index):

ENGR00140507	[IPV_FMPLL] CLKCFG_DIS bit is not available in ESYNCR2	The Bit "CLKCFG_DIS" is not available on "ESYNCR2" register for FADO, whereas "CLKCFG_DIS" is a configurable parameter in current implementation and being used in the source code
ENGR00140496	[IPV_FMPLL] Clock disable bit not available in ESYNCR2	The Bit "CLKCFG_DIS" is not available in "ESYNCR2" register of ANDORRA, whereas "CLKCFG_DIS" is being used in the source code
ENGR00138297	[IPV_FMPLL] ERFD is initialized with invalid value	1. In "Mcu_FMPLL_Init" function, calculation of ERFD value is Invalid for ERFD = 1(Configured in Tresos) if ((ERFD_value + 2UL) < FMPLL_ESYNCR2_ERFD) { /* increase ERFD value to ERFD + 2 to ensure clock frequency doesn't go beyond valid range */ ERFD_value = ERFD_value + 2UL; } Value for ERFD will be 0 in configuration file, and in "Mcu_FMPLL_Init" function ERFD will be incremented by 2 while writing to ESYNCR2 register, which is invalid value Reported Component Baseline: BLN_IPV_FMPLL_MCAL_3.0_01.03.00 file: FMPLL_Mcu_LLD.c
ENGR00141514	[IPV_FMPLL] Integrate the review comment updates	Integrate the following updates to mln line (new label BLN_IPV_FMPLL_MCAL_3.0_01.06.00 needed to be created due to previous Andorra release which used the previous IPV label): 1. IPV_FMPLL:- File – FMPLL_Mcu_LLD.c Development Branch name – dev_engr138297_b35104_mcu_invalid_init_of_ERFD CR No. – ENGR00138297 2. IPV_FMPLL:- File – FMPLL_Mcu_LLD.c Development Branch name – dev_engr138108_b35104_code_review_fixing CR No. – ENGR00138108
ENGR00140483	[IPV_FMPLL] MCU_Pll_WAIT_FMLOCK status reached also when frequency modulation disable	Mcu_GetPllStatus function configures MCU_Pll_WAIT_FMLOCK state also when system clock frequency with frequency modulation is disabled.
ENGR00152940	[IPV_FMPLL] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140232	[IPV_FMPLL] PLL lock problem	When Loss of Clock enable (McuLossOfClockEnable) is configured, the PLL is not getting locked in MCU clock initialization

ENGR00169634	[IPV_FMPLL] Update the file FMPLL_Mcu_LLD.c to remove MISRA rule 12.13 violation	NewWork Description: The check of the MISRA Rules reports the following violation: MISRA 2004 Rule 12.13(Advisory) : increment or decrement combined with another operator It is due to the following statements in the lines 309 and 325: if(timeout++ >Mcu_Cfg_Ptr->McuTimeout) Expected behavior: MISRA rules should be followed (where applicable) Proposed solution (Optional): The solution could be to replace the following lines: while((REG_READ32(FMPLL_SYNFMMR) & FMPLL_SYNFMMR_BSY) == FMPLL_SYNFMMR_BSY) { if(timeout++ >Mcu_Cfg_Ptr->McuTimeout){ break; } } with: while((REG_READ32(FMPLL_SYNFMMR) & FMPLL_SYNFMMR_BSY) == FMPLL_SYNFMMR_BSY) { if(timeout >Mcu_Cfg_Ptr->McuTimeout){ break; } timeout++; }
ENGR00169686	[IPV_FMPLL] Update the function Mcu_FMPLL_GetPllStatus to remove the direct use of Mcu_Cf	NewWork Classification: (internal task, improvement, feature request) improvement NewWork Description: The check of MISRA rules reports the following violation for the file Mcu_LLD.c: MISRA 2004 Rule 8.8(Required) : object/function previously declared: 'Mcu_Cfg_Ptr' at location line 131, file C:\EB\tresos\2010.a\plugins\Mcu_TS_T2D14M20I0R0\src\FMPLL_Mcu_LLD.c Expected behavior: All MISRA Rules should be followed (where applicable) Proposed solution (Optional): The pointer Mcu_Cfg_Ptr, in the file FMPLL_Mcu_LLD.c, is used only by the function Mcu_FMPLL_GetPllStatus, to get the value of Mcu_Cfg_Ptr->McuTimeout. So a solution could be to remove the reference to this pointer from the function and to pass Mcu_Cfg_Ptr->McuTimeout as a parameter to the function.
ENGR00151464	[IPV_FMPLL]: MISRA Violation comments are missing for some of the files	NewWork Description: [MISRA Violation comments are missing for some of the files] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]

ENGR00151671	[IPV_FMPLL]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00152313	[IPV_FlexCAN] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140226	[IPV_PITRTI] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152932	[IPV_PITRTI] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161169	[IPV_PITRTI] RTI timer period is incorrect if the timer is enabled before TVAL0 is loaded	MPC5510 MCAL v1.4.10FBR2.4.0 / GPT issue report Behaviour of the MCAL: - Timer is disabled - TLVAL0 register for the RTI ist updated. - Timer is enabled - sometimes the value of the TVAL0 ist not the same then the one which was written to TLVAL0. So the old value is counted down in case the timer is enabled, this leads to wrong times... The explanation for that is, that the RTI uses a different clock source than the system, which is explained in the datasheet MPC5510RM Rev. 1 06/2008, chapter 28.3.2.1 + 28.3.2.2 Workaround of the project is to write the loadvalue to the TLVAL0 register before it is done by the MCAL routine. Then it is enough time for the synchronization and the value is correctly loaded to TVAL0. But for future versions the MCAL should make this itself.
ENGR00169547	[IPV_PITRTI]-Update Misra violation for PITRTI	NewWork Description: fixed misra violation for pitrti Fix misra violation for IPV_PITRTI Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00152290	[IPV_SIU] - Code traceability must be checked against Design IDs	According to the process, the code traceability must be checked against design IDs, using "@remarks Implements DPORTxxxx" directives; only the situations when design IDs are not applicable must be traced against Requirement IDs At this moment, the PORT driver traceability is checked against Requirement IDs in almost all cases.
ENGR00151856	[IPV_SIU] Compiler warning in "Siu_LLD.c"	Problem detailed description (how to reproduce it): "C:\work\Projects\Bolero3M\Tresos\plugins\Port_TS_T2D22M0I9R0\src\Siul_LLD.c", line 161: warning #550-D: variable "Port_GPIODirection" was set but never used STATIC VAR(uint16, PORT_VAR) Port_GPIODirection[PAD_16BLOCK_NO]; Preconditions: NA Trigger: NA Observed behavior: Compiler warning When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: No warning at compile time Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): NA
ENGR00140463	[IPV_SIU] Configurable option to use / not use the MCAL MCU to enter Low Power Modes	An additional supplier specific, boolean configuration parameter “EnterLowPowerMode” shall become configurable in the plug-in. The default value of this configuration parameter shall be “TRUE”. If this parameter has been configured to “TRUE”, the function “Mcu_SetMode()“ shall not be impacted and behave as specified. If this parameter has been configured to “FALSE”, the function “Mcu_SetMode()“ shall not perform the transition to any low power modes where the core stops execution.
ENGR00144392	[IPV_SIU] Configurable option to use / not use the MCAL MCU to enter Low Power Modes for MA	Problem detailed description (how to reproduce it): [Modification for ENGR140463 in Siu_Mcu_LLD.c file is not having fix for MAMBA platform] Preconditions: [None] Trigger: [None] Observed behavior: [Missing fix for MAMBA platform] Expected behavior: [ENGR140463 should be fixed for all platform] Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [Add following statements in Mcu_SIU_Init_Halt() function after while (((uint32)REG_READ32(SIU_HLTACK) & (uint32)(ModePtr->SIU_Hlt)) != \ ((uint32)(ModePtr->SIU_Hlt) & (~MCU_HALT_CPU_AND_PLATFORM_CLOCK))); #if (MCU_ENTER_LOW_POWER_MODE == STD_ON) ASM_KEYWORD(" msync"); ASM_KEYWORD(" se_isync"); EXECUTE_WAIT(); #endif]
ENGR00169679	[IPV_SIU] Function prototypes to be mapped	Conformance tests results for the ICU drivers are attached. Please analyze them and fulfill the analysis report. In SIU_Icu_LLD_IRQ.c file local functions are not mapped.

ENGR00138816	[IPV_SIU] Implementation of external hardware trigger source field to select the source for the tri	Please implement the external hardware trigger source field to select the trigger source input for ADC according to the Chapter7 System Integration Unit (SIU) from the latest Fado Reference Manual - Rev. 4 released 01/2011, found at: http://cache.freescale.com/files/32bit/doc/ref_manual/MPC5668xRM.pdf?fsrch=1&sr=2
ENGR00169398	[IPV_SIU] Incorrect memory mapping	Problem detailed description (how to reproduce it): Please check if all drivers' symbols are placed in between START/STOP sections present on MemMap.h file. Usually these sections are the following: drv_START_CONFIG_DATA_UNSPECIFIED drv_STOP_CONFIG_DATA_UNSPECIFIED drv_START_SEC_CODE drv_STOP_SEC_CODE drv_START_SEC_VAR_UNSPECIFIED drv_STOP_SEC_VAR_UNSPECIFIED drv_START_SEC_CONST_UNSPECIFIED drv_STOP_SEC_CONST_UNSPECIFIED The following findings were raised by our clients. Even if the current driver is not recorded in the list below please check the consistency. > The following symbols are missing a correct memory mapping. All of these symbols get assigned to section ".text_vle": > MCAL (Leopard BETA 0.9.1, but also RTM1.0.0) > * Functions: > * CAN_BASE_ADDRS: Not found at all > * DIO_SIUL_ReadChannel(), > DIO_SIUL_WriteChannel(), > DIO_SIUL_ReadPort(), > DIO_SIUL_WritePort(), > DIO_SIUL_ReadChannelGroup(), > DIO_SIUL_WriteChannelGroup(): > + Dio_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Dio_TS_T2D17M0I9R0\include\Siul_LLD_dio.c: Not mapped > + Port_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Port_TS_T2D17M0I9R0\src\Siul_LLD_port.c: Not mapped
ENGR00141542	[IPV_SIU] Integrate the review comment updates	Integrate the following updates to mln line - updates after review (new label BLN_IPV_SIU_MCAL_3.0_01.06.00 needed to be created due to previous Andorra release which used the previous IPV label): In Siu_Icu_LLD_CfgEx.h Error: #ifndef SIU_LLD_EX_H #define SIU_LLD_EX_H Correction: #ifndef SIU_ICU_LLD_EX_H #define SIU_ICU_LLD_EX_H
ENGR00152952	[IPV_SIU] Internal refractoring for fixing misra violation in Mcu_SetMode api	NewWork Description: Mcu_LLD_Apply_Mode is returning a Std_ReturnType and it is not recieved to any parameter (not utilized) Which generatess MISRA 2004 Rule 16.10 violation Expected behavior: MISRA 2004 Rule 16.10 Violation should be fixed Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00139498	[IPV_SIU] Mcu_GetResetReason checks an incosistent reset reason	The function Mcu_SIU_GetResetReason obtains the reset reason by checking the SIU_RSR register where each bit is associated to a reset source. In particular the bit CRS does not exist in the SIU_RSR register although the function reads the status of this bit.
ENGR00152928	[IPV_SIU] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00141068	[IPV_SIU] Siu_Icu_LLD_CfgEx.h wrong header define	Wrong header for Siu_Icu_LLD_CfgEx.h file: please replace the wrong one: #ifndef SIU_ICU_LLD_H #define SIU_ICU_LLD_H by the correct one #ifndef SIU_ICU_LLD_EX_H #define SIU_ICU_LLD_EX_H
ENGR00151283	[IPV_SIU] Siu_LLD_CfgEx.h includes PORT typedef in case of DIO driver	Siu_LLD_CfgEx.h includes PORT typedef in case of DIO driver
ENGR00139134	[IPV_SIU] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.

ENGR00132343	[IPV_SIU] compiling Siu_LLD_dio.c produces warnings (parameter X never used)	Andorra V1.9.0 Siu_LLD_dio.c produces the following warnings: "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 672: warning (dcc:1516): parameter PortId is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 713: warning (dcc:1516): parameter PortId is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 714: warning (dcc:1516): parameter Level is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 746: warning (dcc:1516): parameter ChannelGroupIdPtr is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 780: warning (dcc:1516): parameter ChannelGroupIdPtr is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 781: warning (dcc:1516): parameter Level is never used "C:\EB\tresos\2010a_XPC564XA\plugins\Dio_TS_T2D18M1I9R0\src\Siu_LLD_dio.c", line 566: warning (dcc:1517): function reverse_bits16 is never used
ENGR00169624	[IPV_SIU]- comments or code should not exceed 100 characters	Problem detailed description (how to reproduce it): All the files for Icu driver needs to be check so that it will not exceed 100 characters [...] Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00151673	[IPV_SIU]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00139731	[IPV_STM] GPT - Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).

ENGR00155073	[IPV_STM] General updates for Bolero RTM 3.0.2	NewWork Description: Update all the files for MISRA analysis: Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00152312	[IPV_STM] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161853	[IPV_STM]- Update driver files to implement the workaround for LDRA code coverage bug	NewWork Description: Update GPTdriver files to update the comments before the "#endif" statement. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00160267	[IPV_SWT] Compiler warning in Swt_LLD.c	Problem detailed description (how to reproduce it): compiler warning: * [build] Swt_LLD.o "C:\work\Projects\ACG-4.5_Leopard\Tresos\plugins\Wdg_TS_T2D17M1I0R0\src\Swt_LLD.c", line 159: warning #550-D: variable "Wdg_ServiceMode" was set but never used STATIC VAR(Wdg_ServiceModeType, WDG_VAR) Wdg_ServiceMode; ^ Preconditions: The warning occurs only if WDG_KEYED_SERVICE_USED is not defined. Trigger: Compile when WDG_KEYED_SERVICE_USED is not defined Observed behavior: compiler warning When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: No warning (see cPRD on compiler warning treatment) Proposed solution (Optional): NA
ENGR00139347	[IPV_SWT] Leopard RTM - Code Review	Please review the drivers against code review checklist (attached) and fix the findings
ENGR00139348	[IPV_SWT] Remove compiler warnings	The following warnings are reported during compilation using Windriver DiabData 5.8.0.0: ~/Swt_LLD.h:176: warning: file does not end in newline Package: Wdg MPC56xxL Found in version: MCAL3.0_BETA_0.9.2

ENGR00152305	[IPV_eMIOS] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00152938	[IPV_eMIOS] Move to PCLint version 9.00F for GPT	NewWork Description: ENGR00152305 addressed misra fixings only for ICU and PWM related files Hence emios GPT files have to be addressed [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00151513	[LINIF] Update the stubs, to no longer export any BASE headers	NewWork Description: - Update the stubs, to no longer export any BASE headers Expected behavior: - No BASE headers are visible by including the stubs headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00155505	[LIN] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142210	[LIN] Code Coverage analysis doesn't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]

ENGR00160957	[LIN] Configuration variable naming in generated code	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.
ENGR00162606	[LIN] DMA transfer not functioning for LIN Rx	Problem detailed description (how to reproduce it): Please investigate the below issue from customer and find why LIN DMA handler is no triggered when slave data is sent. The customer are using MCAL 3.0 version 2.0.0RTM. They have installed HF2, which is already addressing a DMA issue. So now the DMA is working only in transmission, not in reception. Here is their e-mail describing the problem: I had tried DMA LIN, as what I mentioned in previous email, set break point at DMA TX handler (Lin_InterruptFrame_eSci_A_Tx, vector#: 29), debugger stoped when Lin_Send_Header() called. Next, I removed the TX DMA break point and set break point at DMA RX Handler (Lin_InterruptFrame_eSci_A_Rx, vector#: 30), it never breaked even my slave device responded. Then, I unchecked the DMA config in LIN module (in Tresos), and changed LIN to INT mode, set LIN interrupt handler (Lin_LLD_InterruptHandler_ESCI_0, vector#: 146) and tested, ECU (as a master) sent the header and able to break at RX complete state (see figure below), and manage to receive correct LIN slave respond. I am attaching the customer's configuration. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]

ENGR00151488	[LIN] Extended support for Micro Second Bus	NewWork Description: 1) DMA settings container containing the following parameters: - DMA data transfer size - Number of bytes to transfer - Current iteration count - Destination address adjustment - Last destination address adjustment - The half-point interrupt - The end-of-major loop interrupt 2) Extension of the Lin_ChannelConfigType - LinDMAtransferSize - LinDMAnbytesToTransfer - LinDMAiterationCnt - LinDMAdestinationOffset - LinDMAaddressAdjustment - LinDMAInterruptFlags 3) Changes in the Dma_configure_channel function (see chapter 2.4 0f the attached pdf) Requirement source: Customer Request
ENGR00160577	[LIN] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'

ENGR00162688	[LIN] Incorrect memory mapping	Problem detailed description (how to reproduce it): Please check if all drivers' symbols are placed in between START/STOP sections present on MemMap.h file. Usually these sections are the following: drv_START_CONFIG_DATA_UNSPECIFIED drv_STOP_CONFIG_DATA_UNSPECIFIED drv_START_SEC_CODE drv_STOP_SEC_CODE drv_START_SEC_VAR_UNSPECIFIED drv_STOP_SEC_VAR_UNSPECIFIED drv_START_SEC_CONST_UNSPECIFIED drv_STOP_SEC_CONST_UNSPECIFIED The following findings were raised by our clients. Even if the current driver is not recorded in the list below please check the consistency. > The following symbols are missing a correct memory mapping. All of these symbols get assigned to section ".text_vle": > MCAL (Leopard BETA 0.9.1, but also RTM1.0.0) > * Functions: > * CAN_BASE_ADDRS: Not found at all > * DIO_SIUL_ReadChannel(), > DIO_SIUL_WriteChannel(), > DIO_SIUL_ReadPort(), > DIO_SIUL_WritePort(), > DIO_SIUL_ReadChannelGroup(), > DIO_SIUL_WriteChannelGroup(): > + Dio_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE > + Dio_TS_T2D17M0I9R0\include\Siul_LLD_dio.c: Not mapped > + Port_TS_T2D17M0I9R0\include\Siul_LLD.h: DIO_START_SEC_CODE
ENGR00169767	[LIN] LIN development for Monaco - release RTM 2.0.0	NewWork Description: - Development for Monaco - release RTM 2.0.0 Quality Package: - Generate Stack Size Report - Generate MISRA report - Generate VSMD report - Generate CodeCoverage Report and StaticAnalysis report (LDRA) - Generate Profile report - Generate UM/IM - Generate the Test Specification document Test Report: - Run all tests, generate single tests report and the final SummaryTest Report (.xml and .html files)
ENGR00163400	[LIN] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.

ENGR00169966	[MCU] Add MCU Quality report for MONACO 1.5M RTM 2.0.0	NewWork Description: Test Report, traceability matrix, code coverage metrics and MISRA logs must be provided. Configuration compliancy shall be verified with VSMD check and AMDC tool. The quality package documents should be provided. Expected behavior: The quality package documents and the results of the toolchains should be available with the mcu driver.
ENGR00155507	[MCU] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142211	[MCU] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]
ENGR00142147	[MCU] FLASH wait states setting for 96+ MHz clock	Found Andorra BETA 1.9.0 We found that the code generation of the flash wait states for APC, WWSC and RWSC seems to be wrong or even erroneous. The Generator Macro "MCUFlashclkdep0" seems to manipulate the user-settings for APC, WWSC, and RWSC. E.g. when the user sets 7, it pulls them to zero and so on. A value of 7 is supposed to disable address pipelining (tbd). Please review the documentation and justify each automatic resetting of these values. Please also review if these automatic adjustments are correct. Please also find the attached screen shot with recommended settings for some certain frequencies. The generator seems not to exploit the full speed of the flash in this code snippet: [!ELSE!] [!IF "($F>0) and ($F<=30000000)"!] [!VAR "tmp_B02APC"="0"!] [!ELSEIF "($F>30000000) and ($F<=60000000)"!] [!VAR "tmp_B02APC"="1"!] [!ELSEIF "($F>60000000) and ($F<=90000000)"!] [!VAR "tmp_B02APC"="2"!] [!ELSEIF "($F>90000000) and ($F<=100000000)"!] [!VAR "tmp_B02APC"="3"!] [!ELSEIF "($F>100000000) and ($F<=150000000)"!] [!VAR "tmp_B02APC"="4"!] [!ENDIF!] [!ENDIF!] Crossing frequencies should be 30, 60, 90, 120 and 153 MHz rather than 30, 60, 90, 100 and 150 MHz according to the

ENGR00169695	[MCU] Update the calls to the function Mcu_FMPLL_GetPllStatus to pass timeout as a paramete	NewWork Classification: (internal task, improvement, feature request) improvement NewWork Description: The check of MISRA rules reports the following violation for the file Mcu_LLD.c: MISRA 2004 Rule 8.8(Required) : object/function previously declared: 'Mcu_Cfg_Ptr' at location line 131, file C:\EB\tresos\2010.a\plugins\Mcu_TS_T2D14M20I0R0\src\FMPLL_Mcu_LLD.c Expected behavior: All MISRA Rules should be followed (where applicable) Proposed solution (Optional): The pointer Mcu_Cfg_Ptr, in the file FMPLL_Mcu_LLD.c, is used only by the function Mcu_FMPLL_GetPllStatus, to get the value of Mcu_Cfg_Ptr->McuTimeout. So a solution could be to remove the reference to this pointer from the function and to pass Mcu_Cfg_Ptr->McuTimeout as a parameter to the function from the file Mcu_LLD.c.
ENGR00169467	[MCU] User manual should be updated in the list of return types of Mcu_GetResetReason	NewWork Classification: Improvement NewWork Description: [MCU] User manual should be updated in the list of return types of Mcu_GetResetReason. In the Mcu_GetResetReason function, should be removed the list of return values and should be added a reference to Mcu_ResetType that is specific for each platform. So, should be removed all the @retval and just put: * @return One of the possible reset reason defined in Mcu_ResetType. * * @see Mcu_ResetType Moreover, should be updated the description of MCU_POWER_ON_RESET in Mcu_ResetType table as follow: MCU_POWER_ON_RESET --> Power on reset event. instead of: MCU_POWER_ON_RESET --> Power on event. Expected behavior: User Manual updated Requirement source: NA
ENGR00163389	[MCU] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.
ENGR00118304	[MemIf] - move platform independent files to generic folder	Move all platform independent files to generic folder. Create symlinks to specific folder for MemIf.mak file.

ENGR00152588	[PORT] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00155509	[PORT] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142011	[PORT] PCR 145 was out of range at the PORT plug-in at Monaco MCAL 1.9.0	The PCR 145 is out of the range of possibile connfigurable PORT
ENGR00170352	[PORT] PORT development for Monaco RTM 2.0.0	NewWork Description: - Development for Monaco RTM 2.0.0. Quality Package: - Generate Stack Size Report - Generate MISRA report - Generate VSMD report - Generate CodeCoverage Report and StaticAnalysis report (LDRA) - Generate Profile report - Generate UM/IM - Generate the Test Specification document Test Report: - Run all tests, generate single tests report and the final SummaryTest Report (.xml and .html files)
ENGR00158820	[PORT] Pins PCR231,PCR232 cannot be configured	MPC563XM_MCAL3.0_BETA_HF5_1.9.0 When the pins PCR231, PCR232 are configured, then there is reported the following error: Value "231" of node "/AUTOSAR/TOP-LEVEL-PACKAGES/Port/ELEMENTS/Port/PortConfigSet/PortConfigSet_0/PortContainer/PortContainer_0/PortPin/PortPin_0/PortPinPcr" is out of range: For input string: "Invalid Pin number for current package" This is not correct since as per HW reference manual both pins PCR231,232 support GPIO mode and also other modes. Please review all modes of all pins as well.
ENGR00151872	[PORT] Virtual IO support for Micro Second Bus	NewWork Description: To assign the eTPU/eMIOS signal to the SPI frame, we use virtual general purpose outputs. So we need to have both the MCAL & Tresos able to configure the relative PCR. The PCR numbers are 350 to 413 Expected behavior: Virtual IO management Requirement source: Customer Request
ENGR00163398	[PORT] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.

ENGR00155502	[PWM] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00162929	[PWM] General updates for MONACO RTM 2.0.0	NewWork Description: Update all the files related to below mentioned release activities: AMDC / VSMD reports generation/review MISRA checks UM/IM updating Code Size/Stack Usage reports LDRA Reports generation Traceability matrix Profiling report generation EPD test report Summary reports Review checklist (code, design test plan, requirements) Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00143509	[PWM] In traceability matrix 'sign off' details not updated	In traceability matrix 'sign off' details are not updated (Name, Date).
ENGR00163408	[PWM] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.
ENGR00170346	[PWM]-Update Code template file to support Epd testing	Problem detailed description If Modeselct is OPWFMB, PwmOffset is not supported. if Modeselct is OPWMB, PwmPrescaler and PwmPrescaler_alternate is not supported. Hence the existance of the above mentioned parameters is dependent on the mode selected. Expected behavior: Check has to be implemented in code templates for parameters PwmOffset, PwmPrescaler and PwmPrescaler_alternate Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): [...]

ENGR00162828	[RESOURCE] General updates for Monaco RTM 2.0.0	NewWork Description: Update Traceability matrix sheet and VSMD/AMDC report for Monaco RTM 2.0.0 Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00120171	[SAMPLE_APP] Create Release check application for Leopard Beta 0.9.0 release	Test Release check application for Leopard Beta 0.9.0 release
ENGR00122196	[SAMPLE_APP] Create Release check application for Leopard Beta 0.9.1 release	Create Release check application for Leopard Beta 0.9.1 release
ENGR00132796	[SAMPLE_APP] Port Sample Application Release Check on BOLERO3M EVB	Port Sample Application Release Check on BOLERO3M EVB
ENGR00118883	[SAMPLE_APP] Update for Leopard 0.8.1 P2	Update [SAMPLE_APP] for Leopard 0.8.1 P2 - use the latest plugins to test
ENGR00138974	[SAMPLE_APP] Update integration application with OS for MPC563XM	Update integration application with OS for MPC563XM
ENGR00139914	[SAMPLE_APP] Update release Check application	Run release check application.
ENGR00117955	[SAMPLE_APP]: Create Sample application release check for BLN_MCAL_3.0_LEOPARD_EAR	Create Sample application release check for BLN_MCAL_3.0_LEOPARD_EAR_0.8.1
ENGR00118693	[SAMPLE_APP]: Create Sample application release check for BLN_MCAL_3.0_LEOPARD_EAR	Create Sample application release check for BLN_MCAL_3.0_LEOPARD_EAR_0.8.1_P1
ENGR00117359	[SAMPLE_APP]: Port Sample application release check to Leopard	Port Sample application release check to Leopard.
ENGR00140553	[SARELCHK] Create Release Check Application for XPC56XXA_MCAL3.0_2.0.0_RTM Release	Create Release Check for Andorra RTM.
ENGR00138970	[SARELCHK] Create Release Check Application for XPC56XXL_MCAL3.0_1.0.0_RTM Release	Create Release Check Application for XPC56XXL_MCAL3.0_1.0.0_RTM Release
ENGR00153866	[SARELCHK] Create Release Check Application for XPC56XXL_MCAL3.0_RTM_1.0.1 Release	NewWork Description: Create Release Check Application for XPC56XXL_MCAL3.0_RTM_1.0.1 Release
ENGR00133841	[SARELCHK] Create Release check application for XPC56XXL_MCAL3.0_0.9.2_Beta release	Create Release check application for XPC56XXL_MCAL3.0_0.9.2_Beta release
ENGR00142292	[SARELCHK] ST: Create Release Check Application for XPC56XXK_MCAL3.0_0.9.1_BETA Rel	NewWork Description: […] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00162348	[SARELCHK] Update the Release Check Application for Bolero3M Beta 0.9.1	NewWork Description: Update the Release Check Application for Bolero3M Beta 0.9.1 Expected behavior: Compile successfully all MCAL APIs included

ENGR00158928	[SCHM] Replace tabs by spaces	Problem detailed description (how to reproduce it): There are tabs in some SCHM files - to be replaced by spaces Preconditions: NA Trigger: NA Observed behavior: NA When can it be observed? (at configuration time, at runtime, at compile time?) During tab checks Expected behavior: No tabs Proposed solution (Optional): Replace by spaces
ENGR00139852	[SPI] - Test patterns shall be kept generic for all patterns	All test patterns shall be made generic. Currently some test patterns contains Monaco specific tests. -all jobs using DSPI2 will be mapped on DSPI1 (or DSPI0, if needed) but the names will be conserved; a note in the Test Plan or in the TP.c files shall indicate the naming inconsistence for Monaco -if differences in behavior (TP14, for ex), conditional code will be used according to the number of DSPI units (using SPI_MAX_HWUNIT define) -if existing test cases can not work for Monaco according to the existing configuration, new test patterns will be defines (as for TP003_TC032)
ENGR00155503	[SPI] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00142212	[SPI] Code Coverage analysis don't reach the specified threshold (> 90%)	Problem detailed description (how to reproduce it): [...] Preconditions: [...] 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): [...]
ENGR00160961	[SPI] Configuration variable naming in generated code	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.

ENGR00160693	[SPI] Function prototypes to be mapped	Conformance tests results for the SPI drivers are attached. Please analyze them and fulfill the analysis report. 1. In Spi.c file local and global functions are not mapped. 2. Memory mapping is missing for Dspi_LLD.c and Spi.h file. 3. In Dspi_LLD.h file SPI_START_CONFIG_DATA_UNSPECIFIED is not closed properly.
ENGR00162930	[SPI] General updates for MONACO RTM 2.0.0	NewWork Description: Update all the files related to below mentioned release activities: AMDC / VSMD reports generation/review MISRA checks UM/IM updating Code Size/Stack Usage reports LDRA Reports generation Traceability matrix Profiling report generation EPD test report Summary reports Review checklist (code, design test plan, requirements) Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00139511	[SPI] Timed Serial Bus (TSB) configuration to support Micro Second Bus	It is requested the implementation of a not-Autosar API to support the configuartion of the TSB in the "DSPI DSI Configuration Register (DSPI_DSICR)" to enable user to access the Micro Second Bus functionality. This feature is supported only by DSPI peripheral implemented in Power Train platforms (Monaco and Andorra). The chapter 25.5.9 of the attached reference manual describes the HW features of the TSB. Here below the SW sequence requested by the cvustomer to access the TSB functionality: In order to enable the MSC mode we need the following configurations: 1- DSICR1 register (DSPI_BASE + 0xD0) Full register configuration 2- MCR register: set field DCONF for CSI mode (not SPI) The other MCR configuration should be same as SPI: * master mode * Tx fifo enable * Rx fifo disable * continuous clock mode CONT_SCKE 3- CTAR[x] registers for frame definition & baudrate Then the SPI will be set and when started (Halt bit cleared in MCR register) the transmission of data is automatic and continuous After by SW some command frame must be sent For this it is enough just to write to the PUSHR register like this void SPI_B_CMD(vuint32_t CMD) { DSPI_B.PUSHR.R = CMD; }

ENGR00170008	[SPI] To fix VSMD error in Monaco	NewWork Classification: (internal task, improvement, feature request) [...] VSMD error needs to be fixed[…] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00143503	[SPI] code coverage is less than 90%	NewWork Description: Add test cases to increase the code coverage more than 90% Expected behavior: Code coverage > 90% Requirement source: The code coverage is less than 90% in Pictus 2.9.0 release.
ENGR00120738	[WDGIF] Move WdgIf to independent branch	Move WdgIf on autosar branch (independent of the platform)
ENGR00155508	[WDG] COSMIC compiler support is requested for all the JDP devices	NewWork Description: Support of the COSMIC compiler v4.3.1 Expected behavior: All the drivers must be compiled and tested with the requested compiler Requirement source: gMRD
ENGR00163665	[WDG] Compile error in Wdg_Irq.c	NewWork Description: - Wdg_Irq.c doesn't compile with latest DEM header - Types definition shall be visible before including Dem.h in Wdg_Irq.c Expected behavior: - MCAL Driver code shall not rely on the includes available in stub headers Requirement source: - Defect preventive action plan Proposed solution (Optional): - NA
ENGR00160963	[WDG] Configuration variable naming in generated code	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.
ENGR00163393	[WDG] remove Driver Statistics chapter from UM	NewWork Description: Remove the Driver Statistics chapter from the User manuals Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): This chapter violates all tool licenses that we have. We are not allowed to provide tool compressions.

ENGR00170292	[WDG] test suite development and quality package update for Monaco RTM 2.0.0	NewWork Description: - Development for Monaco RTM 2.0.0. Quality Package: - Generate Stack Size Report - Generate MISRA report - Generate VSMD report - Generate CodeCoverage Report and StaticAnalysis report (LDRA) - Generate Profile report - Generate UM/IM - Generate the Test Specification document Test Report: - Run all tests, generate single tests report and the final SummaryTest Report (.xml and .html files)
ENGR00141148	[ADC] Add tests for 2 or more IP instances on the same platform	In the case an IP has several instances on the same platform (e.g. 2 CAN controllers, or SPI ones), a test should be added to check that there are no interferences when both are running on the same app (see the CAN clone for an example of issue that went undetected until the customer found it).
ENGR00133901	[ADC] Start of a Software triggered conversion should be possible in parallel to a CTU triggered	The original CR ENGR131989 was cloned to treat the 2 requests separately. This clone handles part 1: 1. Start of a Software triggered conversion should be possible in parallel to a CTU triggered group conversion.
ENGR00140289	[ADC] Start of a Software triggered conversion should be possible in parallel to a CTU triggered	Actually it's only possible to have HW and SW conversions in parallel when: 0) AdcPriorityImplementation = ADC_PRIORITY_NONE and AdcEnableQueuing = true => OK (see ENGR133901). Moreover, also the next cases must be possible: 1) - AdcPriorityImplementation != ADC_PRIORITY_NONE and AdcEnableQueuing = true. 2) - AdcPriorityImplementation = ADC_PRIORITY_NONE and AdcEnableQueuing = false. So all cases 0), 1),2) must be considered.
ENGR00133284	[ADC] - Incorrect MACRO values are generated when more than one AdcConfigSets are configu	1. #define ADC_QUEUE_MAX_DEPTH_MAX [!"AdcConfigSet/*/AdcGeneric/AdcPriorityQueueMaxDepth"!]U As per the current implementation the ADC_QUEUE_MAX_DEPTH_MAX generates the configured value for the 'AdcPriorityQueueMaxDepth' only from the first AdcConfigSet. While it should check all the configured AdcConfigSet and the maximum queue depth configured across all the configsets should be generated. 2. #define ADC_MAX_GROUPS [!"num:i(count(AdcConfigSet/*/AdcHwUnit/*/AdcGroup/*))"!]U As per the current implementation the ADC_MAX_GROUPS generates the number of group configured for all the configset. While it should check all the configured AdcConfigSet and the maximum number of groups configured for the configset(which is having the highest configured group) should be generated. 3. #define ADC_MAX_GROUP_CHANNELS [!"AdcConfigSet/*/AdcGeneric/AdcMaxGroupChannels"!]U As per the current implementation the ADC_MAX_GROUP_CHANNELS generates the configured value for the 'AdcMaxGroupChannels'only from the first AdcConfigSet. While it should check all the configured AdcConfigSet and the maximum number of channels configured for the groups across all the configsets should be generated. 4. /** @brief number of channels in respective groups */ [!VAR "index" = "0"!][!// [!LOOP "AdcConfigSet/*/AdcHwUnit/*/AdcGroup/*"!][!// #define ADC_GROUP_[!"num:i($index)"!]_CHANNELS [!"num:i(count(AdcGroupDefinition/*))"!][!VAR "index" = "$index + 1"!] [!ENDLOOP!][!// The "ADC_GROUP_[!"num:i($index)"!]_CHANNELS" MACRO should generate the number of configured channels for the
ENGR00140742	[ADC] - Wrong definition of the ADC HW Queue depth.	ADC_HW_QUEUE should be defined like the max queue depth configured across all configset defined inside the configuration header file.

ENGR00140915	[ADC] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00151243	[ADC] Adc0MaxGroupChannels parameter is set to a fixed value depending on µC derivate	Adc0MaxGroupChannels parameter is set to a fixed value, 56, 76, and so on, depending on µC derivate (changing it will lead to a configuration error). However the description in the UM is different. 4.3.2.2 Adc0MaxGroupChannels (AdcGeneric) It is the count of Maximum number of channels configured for a group for ADC0. [which is having maximum number of channels compared to other groups channel]. In the RTM 3.0.0 this was also described as the maximum number of channels in a group (not fixed as the maximum number of channels available in the HW unit). Did it's meaning / purpose changed between 3.0.0 RTM and 3.0.1 RTM ?
ENGR00139082	[ADC] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152587	[ADC] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140603	[ADC] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00142502	[ADC] Avoid usage of magic numbers	Some of the variables are initialised with magic values, for example, in Adc.c. line # 1122: VAR(Adc_StatusType, AUTOMATIC) tempReturn = (Adc_StatusType)0; >> here, ADC_IDLE should be used instead of magic number "0".
ENGR00142893	[ADC] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-

ENGR00152494	[ADC] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA
ENGR00139746	[ADC] Compiler warnings regarding Adc_Irq.c	Found: MPC560xB in MCAL3.0 RTM 3.0.1 The following compiler warnings are notified by Windriver DiabData 5.7.0.0: ~/Adc/Adc_Irq.c:1550: warning: file does not end in newline
ENGR00133232	[ADC] Create stack size report	Please create the stack size report for each function.
ENGR00142012	[ADC] DMA initialization for QADC	========== File: EQADC_LLD.c @ plugins\Adc_TS_T2D14M1I9R0\src Function name: EQADC_StartDMAOperation: Line2103 of file { …… DestOffset = (uint32)(sizeof(Adc_ValueGroupType) * (uint32)GroupPtr->NumSamples); …… DlastDataValue = (uint32)((2U * (uint32)((uint32)((uint32)(GroupPtr->NumSamples) - 1U) + ((uint32)((uint32)ChannelsInGrp - 2U) * (uint32)(GroupPtr->NumSamples)))) + (uint32)DestOffset); } Condition: NumSamples = 1, ChannelsInGrp = 16 Result with theory: DlastDataValue = 30, but from Monaco DMA side, the value should be 32 Practical result: the value of the TCD.DADDR was changed, and the sampled value was shift into the buffer with different beginning address each time.
ENGR00152616	[ADC] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00151563	[ADC] Exclusive areas information to be provided as part of the driver documentation	For each exclusive area (EA) please include in the IM the ASR API or ISRs names that call the LLD function containing the EA. If this path is affected by configuration, the dependency will be described (config parameters).
ENGR00139918	[ADC] Generate symbolic names for channels indexes (including group name)	The ADC channel symbolic names, generated in ADC_Cfg.h, cannot be used to access the values of the ADC channels inside the ADC result buffers. Solution: Generate ADC symbolic names, that depend also on the ADC group to which each ADC channel is mapped. The generated symbolic name should be something like #define <ADC_GroupName>_<ADC_ChannelName> "Channel index value", where "Channel index value" is the channel index in the current group.

ENGR00142503	[ADC] Improvement of configuration	Remark: 1) in Tresos Studio, parameter "Group Conversion Type" is configurable with full range for both SW and HW triggered groups, which is not plausible; actually, for HW-triggererd groups, the parameter "Group Conversion Type" can only have value "ADC_CONV_TYPE_NORMAL"; >> please grey filed "Group Conversion Type", in case of HW triggered >> groups 2) in Tresos Studio, the parameter "Group Streaming Buffer Mode" is configurable for all group acces modes, which is not plausible; actually, for "Group Access Mode" = ADC_ACCESS_MODE_SINGLE, the parameter "Group Streaming Buffer Mode" can only have value "ADC_STREAM_BUFFER_LINEAR" >> please grey filed "Group Streaming Buffer Mode", in case of single >> acces mode
ENGR00160570	[ADC] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'
ENGR00141658	[ADC] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00139917	[ADC] Names for configuration structures of ADC are not generated	The names, given by user in Tresos, to the configuration structures of ADC, are not beeing generated in the ADC configuration files. Such generation would be useful, because this way, we could initialise ADC with the same names as the ones given to the configuration structures (see example in PORT, ICU driver).

ENGR00154790	[ADC] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00141203	[ADC] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Attached to this CR are the templates for Excel and HTML. For Excel files, for the documents that have revision history (ex: Traceability Matrix, EPD test report), use attached "JDP header - with revision history.xlsx" The other 2 attachments will be used by the tools generating the other documents.

ENGR00137982	[ADC] Replace tabulators with spaces	Tabulators indenting the code should be replaced by spaces. The list of files that mix tabs and spaces: ADCDig_LLD.c ADCDig_LLD.h ADCDig_LLD_CfgEx.h Adc.c Adc_Cfg.c Adc_Cfg.h Adc_Irq.c Adc_LLD.h Adc_PBcfg.c ESCI_LLD.c Fee.c Fr.c Gpt_LLD.c Lin.c Lin_Cfg.c Lin_Cfg.h Lin_Irq.c Lin_LLD.c Lin_LLD.h Lin_PBcfg.c Mcu.h Mcu_Cfg.c Mcu_LLD.h Mcu_PBcfg.c PWM_LLD.c Pwm.c Pwm_Cfg.c
ENGR00139945	[ADC] SDD design documents doesn't contain the File structure package.	AUTOSAR_MCAL_ADC_SDD.eap design documents doesn't have the File structure package in the Static view. ADC Configuration shall be renamed as Plugin Builder/ Plugin Configuration to be inline with the design checklist.

ENGR00143344	[ADC] SDD design documents doesn't have the State machine diagram.	Problem detailed description (how to reproduce it): AUTOSAR_MCAL_ADC_SDD.eap design documents doesn't have the State machine diagram present in the Dynamic View. As per the Checklist_for_Design_Review.xls the state machine diagram should be present in the module SDD design documents. Preconditions: 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): AUTOSAR_MCAL_ADC_SDD.eap design documents shall be updated to include the State machine diagram inside the Dynamic View. For State machine diagram please refer the ADC Autosar SWS specification.
ENGR00139872	[ADC] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format. Also the GPT test plan template is attached.
ENGR00142505	[ADC] UM documentation improvements	Remark: 1) please mention that Adc_EnableHardwareTrigger() does set the CTU register 2) please correct the formating errors in this document, see for e.g. AUTOSAR_MCAL_ADC_UM.pdf, bottom of page 34 (brief Function disable the TriggerSource for group selected by 'Group' parameter.details)
ENGR00140833	[ADC] Wrong queuing in case of HW queue full.	When the HW queuing is full, no group can be appended.
ENGR00141787	[ADC] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header The following duplicated copyright must be removed: <!-- ** Copyright 2006-2011 by Freescale Semiconductor Inc and STMicroelectronics ** All rights exclusively reserved for Freescale Semiconductor Inc and STMicroelectronics, ** unless expressly agreed to otherwise. -->
ENGR00140455	[ADC]Reframe the ADC driver code and comments to respect the code review checklist points	Following Code Review checklist points which are (as per the Rules as part of the JDP Coding Rules) not satisfied by most of the IPVault driver files which is used by the ADC driver. 1. Has been used 80 columns maximum for all code sources? [Rule 1.5] Remark for MCAL:Maximum 100 chars are allowed. 2. Has been respected the line width not exceeded the width of the Comment Blocks? [Rule 5.1] Please find the attached updated ADC code review checklist for the same.

ENGR00139459	[ALL_GENERAL] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00138884	[BASE] Add CER on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value (so a DET event should be signalled) or a data corruption occured (so a DEM event should be signalled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00142214	[BASE] COSMIC compiler integration on Pictus platform	The COSMIC compiler should be used for This activity need some modifications in the code because the actual implementation is not supported by the COSMIC compiler:: 1.Some INLINE functions declared after their usage, should be moved in the upper part of the following files: a.Spi.c b.Spi_LLD.c 2.The SchM files contain the following functions that should be replaced as follow: ASM_KEYWORD uint32 Adc_schm_read_msr(void) { mfmsr r3 } Should be replaced as: uint32 Adc_schm_read_msr(void) { ASM_KEYWORD("mfmsr r3"); } 3. The MemMap.h file should be updated adding the #pragma section expected for COSMIC. 4. The following code should be added in Mcal.h, for Cosmic: #define VAR_ALIGN(v, size) #pragma section [aligned##size] \ v; \ #pragma section []

ENGR00138506	[BASE] E_OK macro defined in Std_Types.h is causing a pre-processor error	The E_OK macro defined in Std_Types.h is causing a pre-processor error when trying to integrate the MCAL with a code like: #if (E_OK != 0) line 184: #define E_OK ((uint8)0x00U) Precompiler doesn't recognise type-cast -> error. Why is the constant zero defined in such a complicated way? The Autosar definition should be used instead: #define E_OK 0x00 CE Note: The same must be done for E_NOT_OK
ENGR00151423	[BASE] HF integration	Cosmic HF integration for Monaco
ENGR00141792	[BASE] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00142385	[BUILD_ENV] - verify DERIVATIVE value against allowed derivative list	NewWork Description: BEART configuration should return an error if an unknown string appears in the DERIVATIVE variable (variable that can be set from command line). The value of the DERIVATIVE variable will be set into all test reports, and causes major problems for the scripts creating the summary test reports. The developer must use only a predefined list of derivatives (the ones supported with linker command files & startups by build environment). Expected behavior: Error if derivative value is not part of a predefined list (like 5604B/5607B for Bolero 512k/1.5M, or 5644A/5642A for Andorra 4M/2M Requirement source: N/A - internal refactoring Proposed solution (Optional): The following checks should be added in config_beart.mak file (example created for Andorra). #check allowed derivatives for this release VALID_DERIVATIVE=FALSE #Andorra 4M ifeq ($(DERIVATIVE),5644A) VALID_DERIVATIVE=TRUE endif #Andorra 2M ifeq ($(DERIVATIVE),5642A) VALID_DERIVATIVE=TRUE endif ifeq ($(VALID_DERIVATIVE),FALSE) #bad derivative name received on command line - report error $(shell echo "Unknown derivative received on command line: $(DERIVATIVE)" 1>&2)

ENGR00156654	[BUILD_ENV] A disclaimer to be added in each Driver User Manual	A disclaimer to be added in each Driver User Manual, to notify the user about this limitation Note: This is preventive action derived during defect analysis, for preventing in the future the occurrence of defects like:ENGR00155365 A proposal for the disclaimer is below: ---------- DISCLAIMER: All containers having the symbolic name tag set as true in the Autosar schema will generate defines like: #define <Container_Short_Name> <Container_ID> For this reason it is forbidden to duplicate the name of such containers across the MCAL configuration, or to use names that may trigger other compile issues (e.g. match existing #ifdefs arguments). ----------
ENGR00138981	[BUILD_ENV] Add CER on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value (so a DET event should be signalled) or a data corruption occured (so a DEM event should be signalled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00139097	[BUILD_ENV] Add support for GHS runtime checks	Add support in build environment to run the tests with GHS runtime checks enabled.
ENGR00157544	[BUILD_ENV] Check start-up code alignment	Please check start-up code alignment, for some platforms there is missing code alignment (ASM_KEYWORD(" .align 4")).
ENGR00140215	[BUILD_ENV] Remove duplicated configuration variables	Duplicate configuration variables to be aggregated. config_beart.mak and config_tpb.mak contain duplicated variables: SW_VERSION_MAJOR SW_VERSION_MINOR SW_VERSION_PATCH SW_VERSION_SUFFIX
ENGR00157545	[BUILD_ENV] Test support APIs to be moved from arfrm to JSW vob	NewWork Description: Several drivers test cases are using the test APIs stored to arfrm\testfrm\test\api\ar3.0\... These files were created by the VnV team but now these are no longer supported. As they are still used by MCAL test cases these must be moved under JSW\JSW_MCAL\XPC56xx\dev\test\test folder
ENGR00140240	[BUILD_ENV] Update CW linker/config files to use VLE libraries.	Update CW linker file to use VLE libraries.
ENGR00156867	[BUILD_ENV] Update build_cfg.mak for each toolchain	NewWork Description: Update build_cfg.mak for each toolchain with the following options: CCOPT:= $(CCOPT) -DEU_DO_ON_CASE_END=\{extern\ void\ SchM_Check_$(MODULE)\(void\)\;SchM_Check_$(MODULE)\(\)\;\} -DMCAL_TESTING_ENVIRONMENT Expected behavior: NA Requirement source: Root cause analysis preventive action Proposed solution (Optional): Update build_cfg.mak for each toolchain with the following options: CCOPT:= $(CCOPT) -DEU_DO_ON_CASE_END=\{extern\ void\ SchM_Check_$(MODULE)\(void\)\;SchM_Check_$(MODULE)\(\)\;\} -DMCAL_TESTING_ENVIRONMENT

ENGR00158344	[BUILD_ENV] cPRD - Use release patch version in the standard plugin name	This is related to a customer request for being able to distinguish (by name) between releases that differ by the patch number (e.g. Bolero RTM 3.0.1 and 3.0.0) A possible solution can be to have release patch version number in the standard plugin name: Case 1: Major version number is zero (0) ========== When Major version number of release is 0, then major version number should be left out in standard plugin name e.g. Bolero Beta 0.8.0: TS_T2D13M8I0R0 (currently TS_T2D13M0I8R0) Bolero Beta 0.8.1: TS_T2D13M8I1R0 (currently TS_T2D13M0I8R0) Case 2: Major version number is non-zero (say 3) ========== When Major version number of release is non-zero (say 3), then both major and minor version number should be used in standard plugin name e.g. Bolero RTM 3.0.0: TS_T2D13M30I0R0 (currently TS_T2D13M3I0R0) Bolero RTM 3.0.1: TS_T2D13M30I1R0 (currently TS_T2D13M3I0R0)
ENGR00140453	[CAN] Add tests for 2 or more controllers on the same bus	The current CAN tests do not include test cases for several controllers (on the same chip) on the same bus. Tests need to be added for this scenario (see also ENGR00140135).
ENGR00144390	[CAN] - Implement the usage of RXGMASK, RX14MASK and RX15MASK	NewWork Description: [The registers RXGMASK, RX14MASK and RX15MASK are provided for legacy support and for low cost MCUs that do not have the individual masking per Message Buffer feature.] Expected behavior: [RXGMASK is used as acceptance mask for all Rx MBs, excluding MBs 14–15, which have individual mask registers RX14MASK and RX15MASK respectively. When the FEN bit in MCR is set (FIFO enabled), the RXGMASK also applies to all elements of the ID filter table, except elements 6-7, which have individual masks RX14MASK and RX15MASK respectively.] Requirement source: [ASR SWS] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...)
ENGR00140916	[CAN] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00139083	[CAN] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).

ENGR00142907	[CAN] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-
ENGR00139694	[CAN] CAN Driver HOH Numbering	4.1.2.3 CAN Driver HOH Numbering The CAN Driver configuration must support the HOH numbering as decided in Bugzilla issue #11714: The numbering for HTHs and HRHs together shall be unique and sequential within the driver and start from 0. Restriction: The HRHs and the HTHs shall be grouped, where HRHs come first. Example: HRH0 - 0, HRH1 - 1, HTH0 - 2, HTH1 - 3 This must especially be supported for configurations containing several controllers what in this case means that the HRHs of all controllers are grouped before the HTHs of all controllers (there must not be any HTH of any controller with an object ID lower than that of any HRH configured for this driver). Please also refer to this Autosar bugzilla entry: http://www.autosar.org/bugzilla/show_bug.cgi?id=11714#c23
ENGR00158702	[CAN] CAN UM was not updated to describe the changed functionality	Problem detailed description (how to reproduce it): The CAN driver behavior for multiplex transmissions was changed during other changes done for ENGR00144127, however the UM was not updated to describe how it should be used. Preconditions: NA Trigger: NA Observed behavior: Code and UM do not match When can it be observed? (at configuration time, at runtime, at compile time?) Runtime Expected behavior: The functionality changes from ENGR00144127 should be also matched by updating the UM Proposed solution (Optional): See UM description of CanMultiplexedTransmission in ASR 4.0, or make sure the following info is clarified in the UM: When “Multiplex Transmission” is ON, several TRANSMIT HardwareObjects can have the same CanObjectID. But they must belong to the same controller. When calling Can_Write function, the parameter Hth is equal to the CanObjectID of the HardwareObject wanted to be used for transmission. “Can Multiplex Transmission” means that the driver will look for a free MessageBuffer in the range of the Hth given by parameter. In the configuration provided, every MB has a different CanObjectID, so the driver has no other MBs to look for. It will use only the MB provided by parameter in Can_Write function.

ENGR00160047	[CAN] CAN second controller not functional when the multiplex transmission is on	Problem detailed description (how to reproduce it): The CAN driver behavior for multiplex transmissions was changed during other changes done for ENGR00144127. By this change, the second CAN controller seems to no longer work. Preconditions: Multiplex transmission on first controller Trigger: NA Observed behavior: After updating the IDs per the clarifications of the ENGR00144127, the multiplexed transmission works fine now, however the second CAN controller is no longer transmitting any message. The customer tried at run time with a debugger to modify the object Ids on the second controller back to what they were with the “non working config” and in that case transmission is Ok Could you check if such a scenario has been tested? By updating by hand the IDs to be in the space of the second controller, it seems the second cotroller starts transmitting. (so it might be an issue with the ID >31 being assumed as part of controller 2 and <=31 as controller 1. Since by mulitplex transmission several MBs share the same ID, that some MB with ID <31 will be part of controller 2 which seems to break the driver. Runtime Expected behavior: Second controller working Proposed solution (Optional): NA
ENGR00144270	[CAN] Can_LLD.c does not check SW version	Can_LLD.c does not check SW version for the included headers Issue found on AUTOSAR MPC56XXA MCAL3.0 swv: 2.0.0 RTM
ENGR00152648	[CAN] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00138100	[CAN] Fado RTM - Code Review against checklist	Please review the drivers against latest code review checklist - available at: http://compass.freescale.net/livelink/livelink?func=ll&objId=215499504&objAction=browse&viewType=1 Checklist_for_Code_Review_NonSafety(Class_BProjects).xls - version 9 of the file in Compass ...and fix the findings. The checklists from Leopard shall be reused (as well as the corrections done on Leopard), Fado code review to be run afterwards only on the delta code between Leopard and Fado.

ENGR00143214	[CAN] Folder structure for test reports not consistent	Problem detailed description (how to reproduce it): The tests added in the zip archive contain one additional folder (not conform with the folder structure). There should be no folder inside the zip file, just test reports. This causes the script to fail to locate the CAN tests for the Summary Test Report Preconditions: None Trigger: None Observed behavior: When unzipping the test reports, another folder layer is created Expected behavior: There should be no folder inside the zip file, just test reports. 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 folder from the zipped test reports.
ENGR00144127	[CAN] HOH check for Tx/Rx order is done on list index and not on Can Object IDs	Problem detailed description (how to reproduce it): On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Error: [...\tresos\plugins\Can_TS_T2D13M3I0R0/generate_PC/include/Can_Cfg.h:351]: The HRH and HTH Ids are defined under two different name-spaces (Refer to CAN326). Example: HRH0-0, HRH1-1, HTH0-2, HTH1-3. The HRHs of all controllers are grouped before the HTHs of all controllers (there must not be any HTH of any controller with an object ID lower than that of any HRH configured for this driver). Note that the object itself does not change, only the position in the list changes. Preconditions: NA Trigger: Configuration time error. Observed behavior: On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Expected behavior: CAN326 refers to IDs, not indexes Proposed solution (Optional): see solution implemented in ENGR00131632

ENGR00156859	[CAN] HOH check for Tx/Rx order is done on list index and not on Can Object IDs	Problem detailed description (how to reproduce it): On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Error: [...\tresos\plugins\Can_TS_T2D13M3I0R0/generate_PC/include/Can_Cfg.h:351]: The HRH and HTH Ids are defined under two different name-spaces (Refer to CAN326). Example: HRH0-0, HRH1-1, HTH0-2, HTH1-3. The HRHs of all controllers are grouped before the HTHs of all controllers (there must not be any HTH of any controller with an object ID lower than that of any HRH configured for this driver). Note that the object itself does not change, only the position in the list changes. Preconditions: NA Trigger: Configuration time error. Observed behavior: On generation, there is an error if any Tx HOH has a list index smaller than a Rx HOH, independent on the real Can Object ID given to those HOHs. Expected behavior: CAN326 refers to IDs, not indexes Proposed solution (Optional): see solution implemented in ENGR00131632
ENGR00153655	[CAN] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>
ENGR00143150	[CAN] Identify implemented workarounds for each platform	We need to identify implemented workarounds for each driver/platform and fulfilling attached file Implemented_Workaround_Review.xlsx. What could help with identifying : 1.User Manual documentation checking 2. Checking of previous errata analysis and CR’s used for workarounds implementation (list in attached file) 3. Checking whether other developers are aware of any “special” code implementation

ENGR00141664	[CAN] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00137616	[CAN] Leopard 1.0.0 - DC100%	Please increase the code coverage to reach the quality goal (DC=100%)
ENGR00136954	[CAN] Leopard RTM - Code Review	Please review the drivers against code review checklist (attached) and fix the findings
ENGR00140149	[CAN] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Templates to be provided for Word, Excel an HTML.
ENGR00142889	[CAN] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).

ENGR00140651	[CAN] Update documentation content	After review some findings need to be fixed in xml documentation files. Integration manual: Bad/incomplete lists of files for CAN dependencies (base,etc) – Chapter 3.2. Why describe “Resource Parameters Configuration”? It’s internal to the driver. “NOTE: GPIO pins used for connection of CAN physical layer have to be properly assigned to the FlexCAN module prior the CAN initialization. API service Can_SetController_Mode(can_controller, CAN_T_START) shall be used for setting the CAN controller to running mode.” Should be done through Port module. Chapter 5.3.2 Macros for Interrupts Add some examples for the generated names. Delete “, at which cycle (10ms, 20ms,...?)” from Chapter 6.1 heading. Chapter 8.1 make sure all tresos params appear in this chapter. User manual: Sort the fields in the configuration parameters chapters as they appear in the plugin. Use the same name from the plugin in the UM. For example, in plugin there is “Can CPU Reference Clock Alternate” in UM is “CanCpuClockRef_Alternate”. I can’t do a match between UM and plugin. So I cannot check that all fields from plugin appear in the UM and the reverse. Chapter 3.9 Wakeup Configuration Do not add the Can.CanConfig.InternalWakeupSupport:STD_ON from the resource. It’s internal to the driver. User must know that on his platform this exists or not. 3.10 Configuration Exporting must not appear in the UM. 3.11 Rx Fifo should be moved to driver design summary. 5.2 Interrupt Handlers already appear in the IM. So should be removed from UM. Chapter 5.1 – move to driver design summary. Chapter 4 The Configuration of Can Bit Timing -> move to driver design summary.
ENGR00139122	[CAN] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.

ENGR00160559	[CAN] Use oscillator info from MCU for FlexCAN	NewWork Description: In the CAN driver (for instance on MPC564xB 0.90 MCAL - but this is true for other MCALs using the same IP), it is necessary to specify manually the oscillator frequency in case FlexCAN is clocked by the oscillator. This information has to be provided in the Mcu driver as well. Why don't we use in the CAN a reference to the MCU driver / McuClockSettingConfig / CrystalFrequency parameter? Indeed, having to change the same information in 2 different modules is not user friendly and can lead to unnecessary debugging sessions... Could this be considered as an improvement for next MCAL releases? Expected behavior: NA Requirement source: Customer Request Proposed solution (Optional): Why don't we use in the CAN a reference to the MCU driver / McuClockSettingConfig / CrystalFrequency parameter?
ENGR00141789	[CAN] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header The following duplicated copyright must be removed: <!-- ** Copyright 2006-2011 by Freescale Semiconductor Inc and STMicroelectronics ** All rights exclusively reserved for Freescale Semiconductor Inc and STMicroelectronics, ** unless expressly agreed to otherwise. -->
ENGR00140917	[DIO] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00139084	[DIO] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152589	[DIO] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140606	[DIO] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.

ENGR00142895	[DIO] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-
ENGR00141777	[DIO] Base.epd and Base.xdm are not required for dio compilation and should be removed from	Chapter 3.2 - Base.epd and Base.xdm are not required for dio compilation and should be removed from the IM
ENGR00152496	[DIO] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA
ENGR00140620	[DIO] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>

ENGR00152624	[DIO] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00143043	[DIO] Fix the design document for Traceability Matrix	During the generation of the traceability matrix the following error message is got: Message: DDIO01004 not found in Design Mapping sheet! Dio.c - 406 Dio.c - 560 DDIO030014 not found in Design Mapping sheet! Dio.xdm - 375 DESIGN002 not found in Design Mapping sheet! Siul_LLD_dio.c - 174 DESIGN001 not found in Design Mapping sheet! Siul_LLD_dio.c - 174
ENGR00142924	[DIO] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.
ENGR00143140	[DIO] Identify implemented workarounds for each platform	We need to identify implemented workarounds for each driver/platform and fulfilling attached file Implemented_Workaround_Review.xlsx. What could help with identifying : 1.User Manual documentation checking 2. Checking of previous errata analysis and CR’s used for workarounds implementation (list in attached file) 3. Checking whether other developers are aware of any “special” code implementation

ENGR00141666	[DIO] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00141684	[DIO] MCAL documentation improvement. General list of improvements.	MCAL documentation improvement. General list of improvements. Applicable to Bolero and Leopard ASR 3.0 MCALs - and all others! list of general improvements for our MCAL documentations. It surely is subject for discussion but these are some generalized points I think our customers really miss. Currently has to read the parameters and then has to read the source code and then has to understand the silicon in order to integrate the driver. This is not only time consuming it also does not give him any certainty obout integrating our code since he does not know whether his interpretation of our code is correct or if the code will even change without notice as there is no specification of how we implemented our driver. See Analysis Report
ENGR00151579	[DIO] NULL pointer check in API Dio_GetVersionInfo(Std_VersionInfoType * pVersionInfo) for the	NULL pointer check in API Dio_GetVersionInfo(Std_VersionInfoType * pVersionInfo) for the parameter 'pVersionInfo' required when DET is ON.

ENGR00139169	[DIO] PCR to pin mapping calculation missing from documentation	Found: Komodo XPC56XXK 3.0 Beta 0.9.0 Customer/CE dialog: Question: I try to configure a LED output using the Port and Dio module. For the Port driver I need to know the PCR number of the corresponding Pin used, which I could find out by: 1.look into the schematics and find out where the LEDs are connected (FLEXPWM0_B0/1 AND FLEXPWM1_A2/3) 2.look into the reference manual to find out which PCR belongs to it (the number which is only described in the text somewhere in brackets behind the alternate I/O row (150 - Ball L16, 151- Ball N15) This is then entered into the Port configuration and I could configure the right output through the drop-down list. Next I wanted to configure the Dio module. For a PortPin, I need to know some 'DioPortId'. According to the Description text, it seemed like it would be the Ball number - but it does not accept Ball L as it says it is not included in the package and there is no Ball I in the Ball grid array (which is described in the Port integration manual as available). Other point is, that the range is always from 0-15, but balls are numbered from 1-17. So how can I find out what I have to enter in order to address a specific PortPin? Answer: If you are using EVB with 257 MAPBGA package (are you?) then LEDs are connected to these PCRs: LED1 = PCR150 = ball L16, A3: flexpwm0_B[1] LED2 = PCR148 LED3 = PCR153 LED4 = PCR 151 = ball N15, flexpwm0_A[2] (as you have correctly mapped the balls->PCR) Ports are grouped into blocks of 16 pins, each pin is address by the offset in the given group. Ex: PCR[a] = m x 16 + n; where a is PCR number,

ENGR00154792	[DIO] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00141205	[DIO] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Attached to this CR are the templates for Excel and HTML. For Excel files, for the documents that have revision history (ex: Traceability Matrix, EPD test report), use attached "JDP header - with revision history.xlsx" The other 2 attachments will be used by the tools generating the other documents.
ENGR00142879	[DIO] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).
ENGR00139873	[DIO] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format.

ENGR00139135	[DIO] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.
ENGR00141793	[DIO] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00155905	[FEE] Typos in Fee Plugins (DeErrorDetect)	On the "General" tab of the plugin, there's improper wording: „Fee Device Error Detect“ instead of „Fee Development Error Detect“
ENGR00140918	[FEE] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00139085	[FEE] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152595	[FEE] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): New test should be created.
ENGR00124847	[FEE] Analyze and correct findings in last Rev.Checklists	There were some findings documented in 'review checklists' now attached in Attach tab. Please analyze and decide if applicable or not.
ENGR00140239	[FEE] Andorra 2.0.0 - Code Coverage100%	Please increase the code coverage to reach the quality goal (Code coverage=100%)

ENGR00142897	[FEE] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-
ENGR00152502	[FEE] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): Will create new test pattern (TP_define_compile) and new test case FEE_TC_MAKGEN.c
ENGR00140625	[FEE] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>

ENGR00151340	[FEE] DataAddrIt is corrupted when corupted block header is parsed	Problem detailed description (how to reproduce it): When the write operation of the block is interrupted in specific time that block header is corrupted containing ECC errors (e.g. ????????_????????_ FFFFFFFF_FFFFFFFF) and this is the last header in the cluster, after initialization data address pointer (DataAddrIt) is corrupted. Preconditions: N/A Trigger: N/A Observed behavior: Write operation fails When can it be observed? (at configuration time, at runtime, at compile time?) runtime Expected behavior: DataAddrIt is corrected before first write or DataAddrIt is not corrupted at all. Preliminary Analysis: When Flash driver reads block header with some error (in this case ECC), Fee_JobErrorNotification() is called. From there, header parse function is called with indication that data in buffer are not valid. Function Fee_DeserializeBlockHdr() detects that the block header is invalid. Then in the function Fee_JobIntScanBlockHdrParse() the corrupted header is skipped by updating the hdrAddrIt to the next header address. But also dataAddrIt is updated according to the value parsed from corrupted header (which is not valid). If the next parsed header is valid, address pointer is updated to the correct value. But if the corrupted header is the last one, address pointer remains corrupted and the next write operation will fail. Proposed solution: 1. Cluster swap shall be initialized, or 2. Blank check shall be performed to find next free address, or
ENGR00138774	[FEE] Documentation issues	-The documentation is missing a lot of information we have provided earlier with Pictus (Explaining Cluster and Block structures, principle functionality, recovery strategies,etc.) -The Integration documentation does not even mention ECC and exception handling. The interfaces and preconditions for that are missing at all.

ENGR00139607	[FEE] ECC in the block data causing incorrect block data pointer computation after swap operatic	Upon finishing the swap operation, the block data pointer (Fee_ClrGrpInfo[Fee_JobIntClrGrpIt].dataAddrIt) is computed incorrectly in Fee_JobIntSwapClrVldDone() if at least one of the copied blocks contains an ECC error in its data. As a result, typically the write operation which triggered the swap operation fails or previously written blank data is overwritten. The reason: during swap operation, the driver is attempting to copy all blocks with valid headers including those with data corrupted by ECC. First, the header of each copied block is written into the new cluster and if an ECC error is detected in the source data then, the swap block operation is aborted and the block is assigned a status FEE_BLOCK_INCONSISTENT. However the space for the data of that inconsistent block remains reserved in the new cluster, which does not correspond with how the block data pointer is computed in the Fee_JobIntSwapClrVldDone(), where the space occupied by inconsistent blocks is ignored because of the following condition: if((Fee_BlockInfo[blockIt].blockStatus == FEE_BLOCK_VALID) || (Fee_BlockConfig[blockIt].immediateData == (boolean) TRUE)) { …. /* Update the block address info */ dataAddrIt -= alignedBlockSize; … } Proposed solution: - block data pointer has to be computed taking into account all blocks occupying data space in the new cluster (including those with the status FEE_BLOCK_INCONSISTENT)
ENGR00139568	[FEE] Empty block instances prepared for immediate data not swapped after FEE init	Issue description: An attempt to write immediate data fails for the following sequence of operations: 1.Empty block instance is prepared with Fee_EraseImmediateBlock() for immediate data block A 2.Reset + FEE initialization is executed 3.The current cluster is filled and swap operation takes place 4.Attempt to write immediate data to block A with Fee_Write() fails The reason is that an empty block instance prepared for immediate data was not copied during swap operation. This is because it was assigned a wrong status during FEE initialization in Fee_DeserializeBlockHdr() - FEE_BLOCK_INVALID instead of FEE_BLOCK_INCONSISTENT, which is assigned originally after Fee_EraseImmediateBlock() is finished. Possible solution: It seems that there is a missing “else” in the Fee_DeserializeBlockHdr(). Instead of if((flagValid == (boolean) TRUE) && (flagInvalid == (boolean) FALSE)) should be else if((flagValid == (boolean) TRUE) && (flagInvalid == (boolean) FALSE))

ENGR00156947	[FEE] Fee driver protection in case FLS driver has disabled some API	following problem occurred during fls integration: It is only possible to activate the synchronous write in the fls config. If asynchronous write is activated there occurs an configuration error in tresos: „Invalid value for node "/AUTOSAR/TOP-LEVEL-PACKAGES/Fls/ELEMENTS/Fls/FlsConfigSet/FlsConfigSet_0/FlsSectorList/FlsSector/FlsSector_9/FlsPageWriteAsynch": When Fls Cancel API is enabled and FEE driver is used, Write Async mode cannot be used." If FlsCancelApi is deactivated then, the corresponding function "Fls_Cancel(void)" will not be generated. This function is missing later in linking process. Fee_Cancel call iy without any guard on compilation errors How is it possible to use asynchronous write with the new mcal?
ENGR00139938	[FEE] Fee test implemented with power supply generator	Fee test implemented with power supply generator
ENGR00157091	[FEE] Fee_Cancel() still calls Fls_Cancel() even when Fls_Cancel API is not active	Problem detailed description (how to reproduce it): if I deactivate the usage of the Fls_Cancel API in the Fls config, the function "Fls_Cancel(void) will not be generated. The problem in this case is, that the the function" Fee_Cancel()" in fee.c uses this missing function. With MCAL 1.0.4 of the Pictus there is no problem, because there is following solution: #if(FLS_CANCEL_API == STD_ON) /* Call Fls_Cancel to cancel the ongoing flash module's job */ Fls_Cancel(); #endif In the fee.c of the bolero this construct is missing. Therefore it is not possible to deactivate the usage of Fls_Cancel API in the Fls config. Could this construct be added to bolero fee.c? Or is it generally not possible anymore for bolero to use async write if Fee and Fls from this release are used? Preconditions: Use Async write for FLS. Use the FEE driver Trigger: Disable the FLS Cancel API Observed behavior: Compiler error When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: Fee Cancel should look at FSL Cancel availability. Proposed solution (Optional): See referred Pictus 1.0.3 solution
ENGR00144271	[FEE] Fee_Cfg.c does not check SW version	Fee_Cfg.c does not check SW version for the included headers Issue found on AUTOSAR MPC56XXA MCAL3.0 swv: 2.0.0 RTM
ENGR00138401	[FEE] Fix robustness code review findings	Fix the findings of the robustness code review described in attached document. In the "Other comments" column, please describe when and by which ticket will be each problem solved.
ENGR00142926	[FEE] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.

ENGR00139958	[FEE] Inconsistancies in Design Document	The AUTOSAR_MCAL_FEE_SDD.eap is not having the following Packages 1. File structure package 2. "BSW" package is not present under MDG Tehnologies
ENGR00139288	[FEE] Inconsistencies in the block headers causing incomplete/obsolete data recovery	Leopard 0.9.2 Issue description: If an inconsistency is detected during FEE initialization in some of the block headers, a swap operation is triggered which copies the latest block instances written before the corrupted instance to a new cluster. The rest of the cluster is ignored even if it contains valid and recoverable block instances. It causes, that some blocks are lost and some recovered with the obsolete data. All blocks in the cluster may be lost in the worst case. One of the potential sources of block header inconsistencies may be the reset or cancel operation occurring during block invalidate or write immediate data operation. Possible solution: FEE initialization: a swap operation should not be triggered upon detection of the first inconsistent block header. The driver should attempt to read at least from the next header position and continue normally with the scanning block headers if a valid header is found there.
ENGR00160573	[FEE] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'
ENGR00141667	[FEE] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".

ENGR00140318	[FEE] Incorrect implementation of the binary search	When function 'Fee_GetBlockIndex' obtains parameter BlockNumber less than the minimum block index used in configuration, the algorithm would run into a negative index with an unsigned value and might have a problem. This scenario can occur when flash contains e.g. valid blocks with indexes 2, 5 and 6. Then from some reason, Fee_Init is called with configuration where block with index 2 is missing.
ENGR00158242	[FEE] Integrate FSL robustness tests into the standard MCAL testing	NewWork Description: Integrate remaining and updated FSL robustness tests into the standard MCAL testing Expected behavior: FEE050 test FEE050 test 2 write_test write_test_2 swap_test immediate_test run for all compiler. Requirement source: Code and .cmm file must be updated for all compiler This suite uses cmm file and is composed by 6 generic file and 2 specific file, therefore every platform should be analyze its specific file (hardware depending). Since last release the FSL test suite was composed by 3 test patterns: TP_01101, TP_01102 and TP_01103. Now, the current FSL test suite is composed by 6 test patterns: TP_04001(FEE050 test), TP_04002(FEE050 test 2), TP_04003(write test), TP_04004(write test 2),TP_04005(swap test), TP_04006(Immediate test). The current FSL test suite runs only for GHS but I will update for all compilers. .

ENGR00152165	[FEE] Issues when "FEE_BLOCK_ALWAYS_AVAILABLE" feature is used	Relevant configuration settings: • Synchronous write operation • Asynchronous erase operation • Fls Erase blank check - not used • Fls write blank check - used • Fls write verify check - not used • FEE_BLOCK_ALWAYS_AVAILABLE = STD_ON Issues: 1.Valid instance of the block is written in the cluster. New instance of this block is being written, but the operation is interrupted by Fee_Cancel() function (it is interrupted after some Fee/Fls_MainFunctions were executed). After Fee_Cancel() we read the block and get “INVALID” or “INCONSISTENT” (depends on time when the write operation was interrupted) block state. It is expected that valid old block is always returned. 2.Invalidated instance of the block is present in the cluster (created using Fee_Invalidate() function). New instance of this block is being written, but it is interrupted by Fee_Cancel() after specific number of Fee/Fls_MainFunctions. After the Fee_Cancel() and also after Fee/Fls initialization the block is read and status “INCONSISTENT” is obtained. In this case, “INVALID” state shall be returned, to keep the information about invalidation operation. 3.There is a valid instance of the block in the cluster. Write operation of the next instance causes a cluster swap, which is interrupted by Fee_Cancel(). Then the block is read, but “INVALID” state is obtained. Issue 3 will be corrected by the follow-up CR ENGR00153256. 4.There is a invalidated block in the cluster (using Fee_Invalidate() function). Write operation of the next instance of this block causes cluster swap, which is interrupted by Fee_Cancel() function (after specific Fee/Fls main functions). After Fee_Cancel() function and also after the Fee/Fls initialization when we try to read the block, we get “INCONSISTENT” block status (we lose information that the block was invalidated) Possible root cases & solutions:
ENGR00152941	[FEE] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00154797	[FEE] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): I will update, with new syntax for Misra warning, the following file:
ENGR00142882	[FEE] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).
ENGR00139874	[FEE] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format.
ENGR00124772	[FEE] VSMD schema file modification	Update of VSMD schema file (XDM) according to findings in ENGR124046, see Analysis tab for details

ENGR00155494	[FEE] Write operation fails when the last block header is corrupted with correct checksum	Problem detailed description (how to reproduce it): This is just robustness issue with low probability of occurrence. If the last block header is corrupted (invalid data addr) in the way that checksum is correct and block header seems ok and DET is OFF, next write operation ends in IVOR exception. The problem is caused in the Fee_JobIntScanBlockHdrParse() by the address update also for unknown FEE blocks. And if the address for such block header points e.g. out of memory, address pointer is corrupted. Preconditions: The last FEE block written to the active cluster seems valid (correct checksum), but address points to wrong place (e.g. 0xFFs). This precondition was reached just by direct write to the memory before test. Trigger: [...] Observed behavior: Write operation ends in IVOR exception. When can it be observed? (at configuration time, at runtime, at compile time?) runtime Expected behavior: Write operation does not fail. Proposed solution (Optional): Maybe address pointer should not be updated in case of unknown ID (because we cannot be sure if the address is correct, safer way is to set address pointer to cause swap) and/or address could be checked for proper value (if the value is in
ENGR00156114	[FEE] Wrong block status after Fee_Cancel()	Problem detailed description (how to reproduce it): "Block Always Available" feature must be OFF. Swap is caused with block which was already successfully written before. When the swap operation is cancelled in very specific time, when the cluster valid flag is already written but the block which caused the swap was not written yet, the block status for this block is wrong. Preconditions: "Block Always Available" feature must be OFF Observed behavior: Block status is wrong When can it be observed? (at configuration time, at runtime, at compile time?) runtime Expected behavior: Block shall be INVALID (not present) after the Fee_Cancel() Proposed solution (Optional): In the Fee_JobIntSwapClrVldDone() function (update of the RAM variables after swap), the block which caused swap shall be marked as INVALID by default (because at this moment it is not in the cluster). Following code could be used before the "for" loop: #if (FEE_BLOCK_ALWAYS_AVAILABLE == STD_OFF) /* invalidate the block causing the swap unless FEE_BLOCK_ALWAYS_AVAILABLE enabled */ if(FEE_NUMBER_OF_BLOCKS > Fee_JobBlockIndex) Fee_BlockInfo[Fee_JobBlockIndex].blockStatus = FEE_BLOCK_INVALID; #endif

ENGR00144274	[FEE] Wrong data are read	Problem detailed description (how to reproduce it): 1. Let A<B<C<D be block IDs. A,B,C – non-immediate, D-immediate block. 2. Reserve space for immediate block D (FEE_EraseImmediateBlock()). 3. Write data to non-immediate block A. 4. Schedule Fee_Write() operation for non-immediate block B. 5. Execute Fee_MainFunction( ) and Fls_MainFunction( ) periodically till first 2 quads of the block B header (ClrID+StartAddress+ClusterSize+Checksum) are written to a memory and Fee_ClrGrpInfo.hdrAddrIt is updated to point to a free space for the next header 6. Execute Fee_Cancel() (and execute Fee/Fls mains till Fee goes to MEMIF_IDLE status) 7. Write a data to immediate block D. 8. Fill the cluster completelly with the instances of the block C and force the cluster swap. 9. Read data from immediate block D – on my side, the job finishes with MEMIF_JOB_OK , but wrong data are returned Preconditions: N/A Trigger: N/A Observed behavior: Wrong data are read Expected behavior: Correct data are read Preliminary Analysis: It seems that this issue has 2 root causes: 1. Only header without data of the inconsistent block B instance is copied to a new cluster. When all blocks are copied to a new cluster, the function Fee_JobIntSwapClrVldDone() is called inside which dataAddrIt is incorrectly updated by the size of the block B. Thus, the blocks processed after block B have a wrong address in the Fee_BlockInfo. dataAddr. 2. Instead of copying the inconsistent block B instance, either the last valid instance or none instance should be copied into a new cluster based on the configuration item defining the compliance with Autosar FEE50 requirement.

ENGR00162275	[FEE] Wrong use of Fee_JobIntClrGrpIt	Problem detailed description (how to reproduce it): 1. There are two clusters and one valid cluster group (not yet initialized), the 0ffset 0x210 is corrupted by ecc (can be any, it will generate cluster swap) 2. After the cluster headers scan, we find the valid cluster at the and Fee_JobIntScanClr is called which set Fee_JobIntScanClr = 1 (out of range for 1 cluster group) 3. Then the valid headers are being scanned for the FEE blocks untill end of initialization. 4. Write blocks until ecc (in our case it try to write the status on ecc location) error notification should be called which set a cluster swap condition on next write (Pictus 3.0.0) This condition is set using Fee_ClrGrpInfo with Out of Range index Fee_JobIntScanClr = 1 Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.)
ENGR00151545	[FEE] [FLS] Integrate FSL robustness tests into the V&V testing	NewWork Description: To integrate all FSL robustness tests into the standard MCAL testing using V&V
ENGR00126310	[FEE] flash block corrupted, consequent read for that flash block will fail	Please comment of the following FAE report: This is a customer request for improvement and not an issue (at least I don't consider it as an issue). After a flash block has been corrupted, any read request for that flash block will fail - also in the case where there are other, outdated/previous instances of that block in the flash available. At the first glance, this looks like correct behavior: if you would compare the EEPROM emulation to the real EEPROM, then you would lose data also in the real EEPROM. However the desired behavior of the customer is to fall back to the most recent previous data instance which should still reside in the flash rather than rejecting the read request. This request also applies to all other MCALs.
ENGR00141753	[FEE] mismatch / missing XDM/XML header	Some modules XDM do not contain a header (e.g.gpt.xdm) or contain a header and a duplication of the copyright (e.g. can.xdm).

ENGR00138845	[FEE] multiple redeclaration of Macro "FEE_CLUSTER_CROUP_CONFIG" for multiple cluster gr	Found: Andorra BETA 1.9.0 The Macro "FEE_CLUSTER_CROUP_CONFIG" is redefined for each additional cluster group (number of groups > 1). Probably, the following lines in the generator file Fee_Cfg.h need to be changed: [!LOOP "FeeClusterGroup/*"!][!// #define FEE_CLUSTER_CROUP_CONFIG \ Should be changed into: #define FEE_CLUSTER_CROUP_CONFIG \ [!LOOP "FeeClusterGroup/*"!][!// Please review if it is really on purpose that all configuration structures in the Fee_cfh.h file shall be placed via the Macro "FEE_CLUSTER_CROUP_CONFIG" directly into the Fee.c file - I don't feel comfortable with that. I would more like to have the declarations in the config header file and the definitions (data allocation) in the implementation config file (fee_cfg.c). Please refer to FEE002 in " AUTOSAR_SWS_Flash_EEPROM_Emulation.pdf", chapter 5.1.2 ("Header file structure").
ENGR00158339	[FEE] some flash locations are corrupted for ageing of the HW	Problem detailed description (how to reproduce it): I created an ECC error at address 0x800200. Then I start writing all blocks, using function Fee_Write(). The first time the block is written to address lower than 0x800200, e.g. 0x800100. Then the test writes the Fee block cyclically forever. However when the Fee reaches the ECC error address 0x800200, then the block is not written at all. The reason is because function Fls_LLD_SectorWrite() performs a blank check using function Fls_LLD_VerifyErase(). Fls_LLD_VerifyErase() return FLS_LLD_E_FAILED, then Fee_Write() also fails forever, (because hdrAddrsesIt is not updated) Preconditions: [...] Trigger: [...] Observed behavior: Fee_Write() also fails forever When can it be observed? (at configuration time, at runtime, at compile time?) at runtime Proposed solution (Optional): So, there are two solutions: In the Fee_JobErrorNotification() will be insert following code: • Our solution (without cluster swap) case FEE_JOB_WRITE: { /* when Fee_Write fails next to next header */ if((Fee_ClrGrpInfo[Fee_JobIntClrGrpIt].dataAddrIt - Fee_ClrGrpInfo[Fee_JobIntClrGrpIt].hdrAddrIt) > (2U *

ENGR00151095	[FEE] update for Misra violation	Problem detailed description (how to reproduce it): In the function Fee_GetBlockIndex at line 834 there is an implicit conversion from an UNSIGNED type (FEE_NUMBER_OF_BLOCKS) to a SIGNED type (high): VAR(sint32, AUTOMATIC) high = FEE_NUMBER_OF_BLOCKS - 1; Trigger: N/A Observed behavior: violation MISRA 2004 Required Rule 10.1 Prohibited Implicit Conversion: Signed versus Unsigned Expected behavior: Explicit Conversion: Proposed solution (Optional): VAR(sint32, AUTOMATIC) high = (sint32)FEE_NUMBER_OF_BLOCKS - 1;
ENGR00153256	[FEE] when "FEE_BLOCK_ALWAYS_AVAILABLE" feature is used BLOCK should not be marke	Problem detailed description (how to reproduce it): when FEE050 is violated ("last valid block always accessible" feature is used), inconsistent blocks should be ignored. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): If there is written valid instance of the block in the cluster and after it invalidated instance is created (new valid instance + Fee_Invalidate()), invalidated instance is ignored by Fee initialization and just valid instance is returned. So we lost information about invalidation process. This issue depends on the "FEE_BLOCK_ALWAYS_AVAILABLE” feature interpretation - if the last valid INSTANCE shall be available or last valid INFORMATION shall be available. Needs to be discussed.
ENGR00153719	[FEE]Compiler warning	In Fee.c file should be removed following warning: variable "clrGrp" was set but never used VAR(uint16, AUTOMATIC) clrGrp =0U; variable "clrGrpIndex" was set but never used VAR(uint8, AUTOMATIC) clrGrpIndex = (uint8)0;

ENGR00153705	[FEE]Read data after FeeCancel are wrong	Problem detailed description (how to reproduce it): • Schedule Fee_Write() for block 1 • Call Fee_MainFunction() • Call Fee_Cancel() • Call Fee_Read() for block 1 Preconditions: FeeBlockAlwaysAvailable = FALSE Observed behavior: at runtime data are wrong Expected behavior: Data must be read Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed Solution I suggest, as possible solution, update Fee.c. Move update operation of Ram variables from Fee_JobWriteHdr() to Fee_JobWriteData() for Normal block and from Fee_JobWriteHdr() to Fee_JobEraseImmediateDone() for Immediate block
ENGR00158518	[FLS] Fls DSI specific defines shall be defined by FSL driver.	NewWork Description: The following defines and types shall be defined by FSL driver: /*** **** DEFINES ***/ #define EXC_UNHANDLED 0 #define EXC_HANDLED_RETRY 1u #define EXC_HANDLED_SKIP 2u #define EXC_HANDLED_STOP 3u /*** **** TYPEDEFS ***/ typedef uint8_least Exc_CompHandlerReturnType; typedef const uint8 * Exc_InstructionAddressType; typedef const void * Exc_DataAddressType; typedef struct { Exc_InstructionAddressType instruction_pt; Exc_DataAddressType data_pt; uint32 syndrome_u32; } Exc_ExceptionDetailsType; typedef const Exc_ExceptionDetailsType *ExceptionDetailsPtrType; /***/ Expected behavior: […]

ENGR00140919	[FLS] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00139086	[FLS] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152594	[FLS] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140610	[FLS] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00124846	[FLS] Analyze and correct findings in last Rev.Checklists	There were some findings documented in 'review checklists' now attached in Attach tab. Please analyze and decide if applicable or not.
ENGR00142896	[FLS] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-

ENGR00139510	[FLS] Block subsequest async requests while another one is pending and DET is off	Concerning the Fls topic (ENGR136892), the answer provided to customer is: This issue is due to a write request being performed by upper layers while an erase request is being processed (When DET is enabled, such request leads to a DET error and new job request being rejected - see FLS030) Now customer is coming back to us arguing that wether DET is enabled or not we should not end up in a situation where DATA flash becomes unusable... "You know as well as I do that the interfaces of the MCAL do not allow to start several asynchronous operations in parallel. The reason is that we have only one GetStatus API with no parameter. We can only check the status on the last operation started. The MCAL must reject any asynchronous operation if another operation is already pending. Otherwise, which status will the MCAL return ? the status of the first operation started, or the status of the last operation started ? This is common sense and does not depend on the DET settings. I really think that we shall agree on this very simple statement and that you shall consider fixing this bug, and checking for all similar issues in the remaining of the MCAL. The implementation was correct on 2.1, why was it modified ? Is there another issue we did not understand?"
ENGR00152501	[FLS] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA
ENGR00152609	[FLS] Compiler warnings to be corrected for Leopard	Problem detailed description (how to reproduce it): We received a new report dealing with some compiler warnings that shall be removed. The type cast in some function calls (Dem_ReportErrorStatus) shall be changed: from "uint16" to "Dem_EventIdType" in the following files: FlexCan_LLD.c Fls.c Modes_LLD.c Monitor_LLD.c We have " FAULT_MCU_E_TIMEOUT_TRANSITION" defined as a "enum" and with the casting you currently have we get warnings: "MCAL\41516132_Modes_LLD.c", line 255: warning (etoa:4188): enumerated type mixed with another type Dem_ReportErrorStatus((uint16) FAULT_MCU_E_TIMEOUT_TRANSITION, DEM_EVENT_STATUS_FAILED); Please make sure that other modules are checked as well. Preconditions: NA Trigger: NA Observed behavior: Compiler warnings When can it be observed? (at configuration time, at runtime, at compile time?) Compile time Expected behavior: No compiler warnings for MCAL code. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The type cast in some function calls (Dem_ReportErrorStatus) shall be changed:

ENGR00143203	[FLS] Correct the duplicated tests	Problem detailed description (how to reproduce it): The drivers test reports contains duplicated test IDs Preconditions: 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): Correct the test IDs and/or remove the duplicated tests. Test Name Test Suite Test Case Master Key TP_02001 TP_02001 TC_02005 fls_TP_02001_TP_02001_TC_02005_cw TP_02001 TP_02001 TC_02005 fls_TP_02001_TP_02001_TC_02005_cw TP_02001 TP_02001 TC_02005 fls_TP_02001_TP_02001_TC_02005_ghs TP_02001 TP_02001 TC_02005 fls_TP_02001_TP_02001_TC_02005_diab
ENGR00152618	[FLS] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00140076	[FLS] Erasetime and Writetime do not have any units	The published parameter FlsWriteTime FlsEraseTime do not have anu unit of measure. No details found within the documentation...
ENGR00139051	[FLS] Erasetime and Writetime not known	What does the published parameter FlsWriteTime with value 5.0E-4 mean? No details found within the documentation... What does the published parameter FlsEraseTime with value 5 mean? No details found within the documentation... What is the erase time, write time and read time when writing and erasing in asynch mode writing 128 bytes at once (FlsMaxWriteNormalMode = 128) reading 128 bytes at once (FlsMaxReadNormalMode = 1024) erasing one 16 kB cluster enabled write blank, write verify, erase blank check (FlsMaxEraseBlankCheck = 1024) calling the Fls_MainFunction() every 10 ms CPU Clock of 80 MHz

ENGR00151718	[FLS] Fls driver not checking validity of addresses passed as parameters	Here is a critical issue requesting a hotfix a.s.a.p. on MPC560xB 3.0.1 MCAL/Fls driver : When the main MCAL functions are executed (Fls_Erase, Fls_Write, Fls_Read, and Fls_Compare) they call a private function named Fls_GetSectorIndexByAddr. This function returns the index of the Flash sector that is impacted by the operation. If the target address is outside the declared Flash range, this private function returns an invalid index. The problem is that this error case is not tested in the MCAL APIs. So if Fls_GetSectorIndexByAddr return an invalid index, this index will be used later on and this generates an access out of the declared table. FAE note: -> I analyzed this and believe (TBC) this is covered only if DET is enabled - In such a case, these functions returns E_NOT_OK and sends FLS_E_PARAM_ADDRESS to DET... the API shall return E_NOT_OK even if DET is disabled... A hotfix is urgently requested for this topic.
ENGR00153485	[FLS] FlsMaxWriteFastMode INVALID check does not work with FeeVirtualPageSize > 8	Problem detailed description (how to reproduce it): Set FeeVirtualPageSize = 16 Preconditions: Fee plug in enabled Trigger: [...] Observed behavior: Write async mode is available When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: Write async mode should be disabled Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): Modify INVALID check into Fls.xdm
ENGR00142326	[FLS] FlsPageSize should have an RANGE type for DFO and DATA	Problem detailed description (how to reproduce it): No error is generated if using pagesize = 8 for DFO Preconditions: [...] Trigger: [...] Observed behavior: [...] Expected behavior: When using FlsPageSize = 8 for DFO flash an error should be triggered in configuration phase

ENGR00153486	[FLS] FlsPageWriteAsynch INVALID check should be dependent on the availability of FEE driver	Problem detailed description (how to reproduce it): The INVALID check for FlsPageWriteAsynch should be valid only if FEE driver is available Preconditions: FEE driver not available Trigger: [...] Observed behavior: FLS driver is not able to set write async mode When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: In case FLS works without FEE there should not be any limitation Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The invalid chech should depend on the availability of FEE driver
ENGR00139147	[FLS] Fls_DsiHandler: which checks are performed by the Dsi_Handler?	This is a generic request for all MCALs using the new codebase: Please specify which checks are performed by the Fls_DsiHandler in order to assure that the ECC exception actually came from an operation issued by the MCAL Fls module. E.g. consider that there might be asynchronous DMA requests or other scenarios which could have led into the ECC Exception. Please specify what you are checking so that the integrator knows what he still needs to check.
ENGR00142404	[FLS] Fls_Erase (async) should work for all kind of Flash on High-address space sectors	NewWork Description: Erase and Write job should work in Async Mode for Standard flash and "Interleaved flash" on High Address space Expected behavior: Async job on High address space works for all kind of Flash Requirement source: […] No cPRD is present, it should be proposed Proposed solution (Optional): [...]
ENGR00152342	[FLS] Fls_Write do not correctly check write end address	Problem detailed description (how to reproduce it): Execute a fls_write untill and of flash Preconditions: Na Trigger: [...] Observed behavior: FLS_E_PARAM_LENGTH is launched by DET When can it be observed? (at configuration time, at runtime, at compile time?) At runtime Expected behavior: Fls_Write should work until end of fLASH Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional):
ENGR00143878	[FLS] Function Cer_ReportError declared implicitly in Fls.c	Fls.c uses a macro from Cer.h but does not include this file. The MCAL Beta version 2.9.0 is also affected.
ENGR00142927	[FLS] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.

ENGR00137800	[FLS] Handling of Flash Lock registers	[FAE]: MCAL 0.9.1 for MPC56xxL Our customer has requested a modification of the Fls driver. Currently Flash Lock registers (i.e. LML, HBL and SLL) have to be set in the user application code before calling Fls_xxx APIs. The customer's request is to merge this step into Fls_Init(), so that the user simply has to call Fls_Init(). [CE]: It appears that every customer wants something else, there was just opposite request recently. TWG topic? Please provide your reasoning for decoupling of function calls for Fls_Init() and Fls_Unlock(). Thanks!
ENGR00138402	[FLS] Implement robustness code review improvement proposals	implement the findings of the robustness code review described in attached document. In the "Other comments" column, please describe when and by which ticket will be each problem solved. The Multiple return statements will not be implemented under this CR, instead these will be adressed in NEW CR ENGR00140170.
ENGR00139956	[FLS] Inconsistancies in Design Document	The following packages are not available 1. File structure package 2. All the Tresos parameters are not documented in Design
ENGR00160574	[FLS] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'
ENGR00141668	[FLS] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".

ENGR00138729	[FLS] Invalid module description because of double node ECU-PARAMETER-DEFINITION	The node ECU-PARAMETER-DEFINITION is listed twice within module description files Fls~.epd. One of them should be removed.
ENGR00142317	[FLS] Linear address not well managed in case of Async write for FlsMaxWriteNormalMode > 8	Problem detailed description (how to reproduce it): Using a configuration with: - Flash blocks not contiguous - FlsMaxWriteNormalMode = 32 - Erase and Write job in ASYNC mode writing 32 across two non contiguous block a writing error is detected Preconditions: [...] Trigger: [...] Observed behavior: Only 8 byte are correctly written by Fls_MainFunction() Expected behavior: 32 byte shopuld be correctly written by Fls_MainFunction() Note: in the “Expected behavior” field, please mention also the requirement source too Proposed solution (Optional): Only in case of Async Write job update Fls_JobSectorIt at the end of Fls_LLD_MainFunction.
ENGR00137236	[FLS] MemMap Section FLS_START_SEC_VAR_UNSPECIFIED not closed	The section FLS_START_SEC_VAR_UNSPECIFIED is used in Fls.c, but FLS_STOP_SEC_VAR_UNSPECIFIED is never called. FLS_START_SEC_VAR_UNSPECIFIED is then called a second time. See lines 592 + 596
ENGR00152946	[FLS] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00154796	[FLS] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]

ENGR00137865	[FLS] System is locked during Fls_LLD_SectorErase (WDG callback implementation)	“What we need for integration is a Fls driver that erase function works either in async mode or sync mode, but with triggering the Watchdog. To solve this issue we already suggested to add a WDG callback function that will be triggered within erase function. This was done in previous versions of the MCAL for Pictus and Bolero. Otherwise we need the a async function for erase.” To note that the mentioned MCAL versions are the 2.1 Bolero and the ST codebase for Pictus. Based on this latest info, I assume you already know what ST has implemented in terms of FLS-WDG for the Pictus ST codebase. Hello, during the integration of the FLS we run in problems when erasing the FLS sectors. After requesting an erase of the sector we will run in a WDG timeout. We need to use the FLs in async mode., so we would expect that there is no lock . The WDG is configured to 20 ms. After debugging the function Fls_LLD_SectorErase we found out that following sequnce within this function takes more than 200 ms: /* back-up PFCR0 or PFCR1 */ Fls_LLD_pfcConfRegValue = Fls_LLD_pfcRegBasePtr[Fls_LLD_pfcConfRegOffset]; /* start internal erase/program sequence */ Fls_LLD_regBasePtr[FLASHMEM_MCR] |= MCR_EHV; Fls_LLD_Job = FLS_LLD_JOB_ERASE; Fls_LLD_JobResult = MEMIF_JOB_PENDING; During this time the WDG is not triggered, the system performs a reset. Attached you can find the configurations used. Do we need to configure something else or why does the system take so much time? Please check this issue and give some feedback.
ENGR00155835	[FLS] The parameter "physicalSectorSize" in "Fls_LLD_SectorErase( )" function needs to be gua	The parameter "physicalSectorSize" will only be used by "Fls_LLD_SectorErase( )" function if the precondition "FLS_ERASE_BLANK_CHECK" is turned ON Reported component baseline: BLN_IPV_FLASH_MCAL_3.0_01.09.00 Origin tag: HF_MCAL_3.0_BOLERO_RTM_HF4_3.0.0 File: Fls.c NOTE: this ticket has been cloned after development to correct DB errors
ENGR00139129	[FLS] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.

ENGR00138064	[FLS] Write operation ends with DET	Configuration description: -Async page write mode -FLS_WRITE_BLANK_CHECK turned on -No Immediate data Issue description: Write block with size greater than 40Bytes to the empty cluster which is last in the configuration (no other blocks are written in the cluster yet). Write operation ends with DET error FLS_E_VERIFY_ERASE_FAILED. This failure is caused when Fls driver tries to write bytes 40-47, in the Fls_LLD_VerifyErase(Fls_LLD_targetAddressPtr, Fls_ConfigPtr->sectorPageSize[Fls_JobSectorIt]) function. It is caused by wrong value in the Fls_JobSectorIt, it is set to value out of the Fls_ConfigPtr->sectorPageSize range, after writing bytes 32-39 in the code: /** @remarks Implements DFLS11204 */ /* Update the Fls_JobAddressIt iterator */ Fls_JobAddrIt += sectorTransferLength; /** @remarks Implements DFLS11206 */ if(Fls_JobAddrIt > Fls_ConfigPtr->sectorEndAddr[Fls_JobSectorIt]) { /* Move on to the next sector */ Fls_JobSectorIt++; } How to invoke the issue: Configure only 2 clusters and 1 block with size 48Bytes. Write block to the flash in cycle and check job result. When write fails, cluster swap occurred. First cluster will be full, second cluster will contain invalid header and at the end of this cluster, only 40 bytes of data will be written (because write failed when writing bytes 40-47). Documentation issues:
ENGR00144272	[FLS] [FEE] Immediate data written after FEE_Cancel() are lost	Problem detailed description (how to reproduce it): 1. Reserve space for immediate block A (FEE_EraseImmediateBlock()). 2. Fill the cluster completelly and schedule Fee_Write() operation to trigger a swap (use non-immediate block B). 3. Execute Fee_MainFunction( ) and Fls_MainFunction( ) fcns periodically till a status is written to a new active cluster 4. Execute Fee_Cancel() (and execute Fee/Fls mains till Fee goes to MEMIF_IDLE status). 5. Write a data to immediate block A. 6. Reinitialize the module. 7. Try to read data from immediate block A – on my side, the read job fails with MEMIF_BLOCK_INVALID result. Preconditions: Asynchronous write mode Trigger: N/A Observed behavior: After initialization written immediate data are lost. The root cause is in step 5 - the immediate data were written to the cluster which is recognized as obsolete after the module re-initialization Expected behavior: After initialization written immediate data are not lost.
ENGR00141754	[FLS] mismatch / missing XDM/XML header	Some modules XDM do not contain a header (e.g.gpt.xdm) or contain a header and a duplication of the copyright (e.g. can.xdm).

ENGR00144273	[FLS][FEE] No data can be written after FEE_Cancel()	Problem detailed description (how to reproduce it): 1. Reserve space for immediate block A (FEE_EraseImmediateBlock()). 2. Schedule Fee_Write() operation of a normal block. 3. Execute Fee_MainFunction( ) and Fls_MainFunction( ) periodically till a first quad (ClrID+StartAddress) is written to a memory. 4. Execute Fee_Cancel() (and execute Fee/Fls mains till Fee goes to MEMIF_IDLE status) 5. (optional) – Write data to an immediate block A 6. Try to write data to any block – on my side, the write job fails with MEMIF_JOB_FAILED result Preconditions: N/A Trigger: N/A Observed behavior: No data can be written. Expected behavior: Data can be written. Preliminary Analysis: The Fee_Cancel() has been issued at the specific time when the part of the header pointed to by the Fee_ClrGrpInfo[]. hdrAddrIt was already written to the memory but the pointer itself had not been updated yet. Thus, the 1st non-immediate block write after FEE_Cancel() attempts to write the header to the location which is already written, which leads to MEMIF_JOB_FAILED .
ENGR00139681	[Fee] One FEE_BLOCK_OVERHEAD is NOT left blank anymore	I have one remark regarding the change done in the Fee_JobWriteHdr(). The original condition /* One FEE_BLOCK_OVERHEAD must be left blank */ if(((uint32) alignedBlockSize + (2U * FEE_BLOCK_OVERHEAD)) > availClrSpace) {...} was changed to /* One FEE_BLOCK_OVERHEAD must be left blank */ if(((uint32) alignedBlockSize + (FEE_BLOCK_OVERHEAD)) > availClrSpace) {...} This change allows to fill the cluster completely in such a way, that the data of the last written block follows immediatelly its header without any blank space between them. It seems a little bit risky to me, as it might potentaily cause that the data of the last written block will be misinterpreted as the next block header, especially when the previous header will be corrupted. Or, please, is there any special need for this change I'm not aware of?
ENGR00139951	[GPT] Inconsistancies in SDD design document	AUTOSAR_MCAL_GPT_SDD.eap design documents doesn't have the File structure package in the Static view. "BSW" package is missing under MDG Tehnologies
ENGR00140920	[GPT] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00141260	[GPT] - update ts_08.mak to remove MCU dependency	the current ts_08.mak includes he path to MCU driver. If this is not present in Tresos, the code compiles. If it is, the test will try compiling also the MCU sources when the MCU configuration is not yet generated.

ENGR00140094	[GPT] :The Maximum counter value is not correct for eMIOS timer channels	For Monaco and Mamba platform Supported maximum counter value is 24 bit (0x00ffffff) for eMIOS timer channel, but the maximum counter value is defined as 20 bit (0xfffff) in Gpt_Cfg.h #define EMIOS_EMIOSCNT_MAX_VALUE (uint32)0x00FFFFFuL Should be replace by #define EMIOS_EMIOSCNT_MAX_VALUE (uint32)0x00FFFFFFuL
ENGR00139287	[GPT] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152598	[GPT] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140141	[GPT] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00142904	[GPT] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-
ENGR00140236	[GPT] Channels are stopped regardless of wake-up configuration	Function Gpt_LLD_SetSleepMode is stopping all channels regardless of wake-up configuration/enablement. FUNC (void, GPT_CODE) Gpt_LLD_SetSleepMode(void) { .... for (channelIndex = GPT_LOOP_INIT; channelIndex < Gpt_Cfg_Ptr->Gpt_Channel_Count; channelIndex++) { .... Gpt_ChannelStatus[channelIndex] = GPT_STATUS_STOPPED; } } However, for channels with wake-up enabled state, the channel should be left running

ENGR00152505	[GPT] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA
ENGR00142930	[GPT] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.
ENGR00141670	[GPT] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00152933	[GPT] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00154800	[GPT] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00140153	[GPT] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Templates to be provided for Word, Excel an HTML.
ENGR00142886	[GPT] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).
ENGR00141416	[GPT] Testing the GPT nonAutoSAR feature	The test pattern 016 is used to test the nonAutoSAR functionality support for channel prescaler setting at runtime. The TP contains 3 tests. For a mistake it is run only test no 1.
ENGR00139125	[GPT] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.

ENGR00141755	[GPT] mismatch / missing XDM/XML header	Some modules XDM do not contain a header (e.g.gpt.xdm) or contain a header and a duplication of the copyright (e.g. can.xdm).
ENGR00155933	[GPT] review of the driver design according to faults in PR 39141739 and to the structure of the H	Problem detailed description (how to reproduce it): [...] Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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): [...]
ENGR00161820	[GPT]- Update driver files to implement the workaround for LDRA code coverage bug	NewWork Description: Update GPTdriver files to update the comments before the "#endif" statement. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140438	[GPT]: The Integration Manual and the User Manual have to be updated.	There are some changes to be implemented in the 2 manuals: User Manual - Ch.2:Introduction: Table 1.2 is not constructed with reference (link). - Ch.3:Driver:Additional Requirements and Deviations: Table Deviation from Requirements is empty. Check if really are/not deviations from Autosar reqs and if not add NA or N/S tyo every cell table. - Ch3:Driver:Function Definitions: APIs doesn't have precondition and postcondition. In which context an API can be called and not called. What is the output of every API and how is changed the driver state. Some APIs description contains Misra Violations as Note - This should not be here. We a special report for this. API that are conditioned by precompile parameters should be explained. Example: Gpt_Cbk_CheckWakeup can be called only if wakeup functionality is enabled or exist on the platform. - Ch3:Driver:Configuration Parameters: Every Parameter should contain a Note for every created table to detail if there are dependencies. Also should explain how is affected the code by setting this parameter on or off. - It is recommended to have special (sub)chapters for important configuration: like wakeup functionality. How can the driver pass from STOP-RUN-SLEEP-WAKEUP, etc...state machine…. - It is recommended to have a chapter about "driver configuration export". Where is declared and exported the "Gpt_ConfigType" strucure. How can the API access it. - It is recommended to have a special chapter about interrupts implementation. How can be interrupts configured or enabled from the plugin, what are the available interrupt handlers and when can be used based driver configuration. Where are interrupts mapped into the vector address table (index). Integration Manual - All the chapter titles should use capital letters. Ex: "Building the driver" -> "Building the Driver" - Ch.2: Introduction

ENGR00161698	[HF] MONACO BETA 1.9.0 HF8 preparation	MONACO BETA 1.9.0 HF7 preparation Released modules LIN SPI PORT Resource
ENGR00151277	[ICU] - Wakeup shift implementation is not tested	Icu_EnableWakeup() API was modified (ENGR00140830) to correct the Wakeup shift as shown below: if (0U != (Icu_Cfg_Ptr->Icu_ChannelConfigPtr[Channel].Icu_ParamValue & (uint32)ICU_WAKEUP_MASK)) But no test was added to test this scenario. This can be tested only when ICU_DEV_ERROR_DETECT switch is STD_OFF. Preconditions: [...] 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): [...]
ENGR00140921	[ICU] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00142903	[ICU] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-

ENGR00151410	[ICU] Can not start timestamp when channel symbolic names are not configured by ascding orde	Problem detailed description (how to reproduce it): The customer configure 4 channels(ICU_MODE_TIMESTAMP mode) in EB as below. Index ChannelName ChannelID Hardware 0 Channel02 0 ETIMER_0_CH_2 1 Channle04 3 ETIMER_0_CH_4 2 Channel11 2 ETIMER_1_CH_1 3 Channel05 1 ETIMER_0_CH_5 4 Channel14 4 ETIMER_1_CH_4 And In application they start timestamp channel05 and channel04 but this function does not work. This is because channel symbolic name does not reflect the index in configured hardware channel list generated by EB Tresos. The symbolic names are defined as configured in tresos. The customer expect that in case this kind of configuration is not supported, EB Tresos plugin should check and throw error message. They also expect that this kind of defect should be fixed in RTM version. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] 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):
ENGR00133355	[ICU] Configuration data should reside in one segment for module configuration (ICU_START_C	Add the configuration pointers under ICU_START_CONFIG_DATA_UNSPECIFIED not ICU_START_SEC_VAR_UNSPECIFIED

ENGR00160601	[ICU] Delayed measurement of High/Low time	Problem detailed description (how to reproduce it): When using an eMIOS unified channel that supports IPWM mode , to measure ICU_LOW_TIME and ICU_HIGH_TIME the value returned is one period delayed. Preconditions: Channel configured for icu measurement mode Observed behavior: The value returned is one period delayed for above properties measured When can it be observed? (at configuration time, at runtime, at compile time?) At runtime Expected behavior: The measured value should be available when the measurement is complete (on trailin edge) Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The changes suggested in ENGR00155079 involve the code bellow found in Icu_LLD.c in function Icu_LLD_SignalMeasurement. if ((Icu_Cfg_Ptr->Icu_ChannelConfigPtr[numChl].Icu_ParamValue & ICU_EMIOS_UC_MODE_PARAM_MASK) == ((Icu_ParamType)EMIOS_UC_IPWM_MODE << ICU_EMIOS_UC_MODE_PARAM_SHIFT)) { /* Check if a complete signal was acquired */ if (0U == Icu_Int_Counter[msChannel]) { /* Mark that the complete signal was acquired */ Icu_Int_Counter[msChannel] = 1U; } else
ENGR00159690	[ICU] Driver generates compiler warnings when compiled with DIAB 5.8.0.0	Problem detailed description (how to reproduce it): While compiling with DIAB 5.8.0.0, ICU driver generates below compiler warnings. "narrowing or signed-to-unsigned type conversion found: " Preconditions: DIAB 5.8.0.0 compiler version shall be used for building driver. Trigger: [...] Observed behavior: Compiler warning is generated during building driver. When can it be observed? (at configuration time, at runtime, at compile time?) compile time 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 the compiler warning in driver files. If it can not be fixed, update the comments in driver code with proper doxygen tags.
ENGR00151568	[ICU] Exclusive areas information to be provided as part of the driver documentation	For each exclusive area (EA) please include in the IM the ASR API or ISRs names that call the LLD function containing the EA. If this path is affected by configuration, the dependency will be described (config parameters).

ENGR00142929	[ICU] Handling compiler warnings	Based on the TWG decision the following solution shall be implemented on all drivers: 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> [... multiple line comment if needed] */ All previous workarounds which were implemented (e.g. dummy code) shall be removed.
ENGR00153657	[ICU] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>
ENGR00141748	[ICU] Inconsistancies in Design Document	1. File structure package is missing. - add all the dependencies and all the files (also the middle layer files and low level driver file) 2. State machine diagram is missing - build a state machine diagram related to ICU_IDLE and ICU_ACTIVE states of the driver. Please refer to the specification for more details.
ENGR00155948	[ICU] Inconsistancies in Design Document	Detail the sequence diagram – the sequence diagram in ICU design should have been refereeing to the layers from the module = something like Icu_Init is calling Icu_LLD_InitChannel and this function is calling SIU_LLD_InitCHannel or eMIOS_ICu_LLD_Initchannel, depending of the channel type.
ENGR00141671	[ICU] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".

ENGR00160604	[ICU] Noncoherent measurement of duty cycle using eMios in SAIC mode	Found:BLN_MCAL_3.0_BOLERO3M_BETA_0.9.0 May also involve other MCUs with eMIOS Problem detailed description (how to reproduce it): When using an eMIOS unified channel that does not support IPWM mode , to measure DUTTY_CYCLE the value of the active time returned does not corespond with the period that is in. Proposed solution (Optional): The portion of code bellow found in Icu_LLD.c in function Icu_LLD_SignalMeasurement takes care of SAIC duty cycle measurement. else { /* DUTYCYCLE or PERIOD measurement */ if (Icu_Int_Counter[msChannel] == 1U) { Icu_ActivePulseWidth[msChannel] = pulse_width; Icu_Int_Counter[msChannel] = 2U; } else { /* counter = 2 */ Icu_Period[msChannel] = Icu_ActivePulseWidth[msChannel] + pulse_width; /* set to 1 to find active pulse width next time */ Icu_Int_Counter[msChannel] = 1U; Icu_LLD_SetBitChState(numChl, ICU_CHANNEL_STATE_IDLE); } /* store for next time */ Icu_Start[msChannel] = tempA; }

ENGR00154801	[ICU] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00140152	[ICU] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Templates to be provided for Word, Excel an HTML.
ENGR00139124	[ICU] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.

ENGR00140830	[ICU] Wakeup shift bits mismatched	Found: Bolero MPC560xB 3.0.1: Mismatch between: In Icu_Pbcfg.c: CONST(Icu_ChannelConfigType, ICU_CONST) Icu_InitPBChannel_0[5] = { /* ICU_WKP_LIN_RX0 - WKUP_11 */ { (((Icu_ParamType)ICU_WAKEUP_CAPABLE << ICU_WAKEUP_SHIFT) | (ICU_BOTH_EDGES << ICU_EDGE_PARAM_SHIFT)), ICU_MODE_SIGNAL_EDGE_DETECT, 0U, Icu_LINWakeUpNotification, #if defined(ICU_OVERFLOW_NOTIFICATION_API) && (ICU_OVERFLOW_NOTIFICATION_API == STD_ON) NULL_PTR, #endif /* ICU_OVERFLOW_NOTIFICATION_API */ 64U }, with ICU_WAKEUP_SHIFT=31 In Icu_EnableWakeup: if (0U != (Icu_Cfg_Ptr->Icu_ChannelConfigPtr[Channel].Icu_ParamValue & ICU_WAKEUP_CAPABLE)) without ICU_WAKEUP_SHIFT! That doesn't match . Because of this the wakeup bit in the field Icu_ChannelState will not be set and if the function Icu_SetMode(ICU_MODE_SLEEP) is called all wakeup IR are deactivated again. Customer reported this on MCAL 3.0.0, I checked MCAL 3.0.1 , this is also valid.
ENGR00156653	[ICU] Wrong computation of PeriodTime and ActiveTime when EMIOS counter overflows	Wrong PeriodTime or ActiveTime is returned by Icu_GetDutyCycleValues in case the measurement is done when the internal EMIOS channel counter is overflowing from value 0xFFFFFF to 0. In this case the next captured time value is less than the previous one and the following computations don’t work as expected (done with 32-bit variables), in Icu_LLD_SignalMeasurement function: Icu_ActivePulseWidth[msChannel] = (Icu_ValueType)(tempA - tempB); and Icu_Period[msChannel] = (Icu_ValueType)tempA - Icu_Start[msChannel]; Both subtractions in above statements should be implemented as modulo 24-bit as the EMIOS internal counter is 24-bit wide.
ENGR00141756	[ICU] mismatch / missing XDM/XML header	Some modules XDM do not contain a header (e.g.gpt.xdm) or contain a header and a duplication of the copyright (e.g. can.xdm).
ENGR00161819	[ICU]- Update driver files to implement the workaround for LDRA code coverage bug	NewWork Description: Update ICU driver files to update the comments before the "#endif" statement. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00159913	[ICU]-Incorrect typecasting for the macro	In file Icu_LLD.c Macro ICU_IRQ_CHANNEL is typecasted with "Icu_ChannelType" but correct method is it should be typecasted with "SIUL_Icu_IRQ_ChannelType" to have better readability Macro ICU_WKPU_CHANNEL should be typecasted with "WKPU_Icu_ChannelType" to avoid diab compiler warnings
ENGR00161496	[ICU]-Update Misra violation for ICU	NewWork Description: fixed misra violation for emiosr Fix misra violation for Icu_LLD.h […] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00141328	[LIN] Configuration enhancement to support Micro Second Bus upstream channel	The LIN implementation on the eSCI peripheral shall support the capability to configure the serial channel as upstream channel for th Micro Second Bus. The eSCI is used as receiver and the supported frames format shal be either the 16-bit or the 14bit MSC upstream frames.
ENGR00140922	[LIN] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00139090	[LIN] Add Catastrophic Errors Recovery (CER) on unreacheable default cases	There are switches for which the default case cannot be reached (the case variable is part of a finite choice list). For such cases, reaching the default case means an out of range value or a data corruption occured (so a CER event should be signaled). NOTE: Please read analysis for description. DET or DEM should not be used - but a new concept - Catastrophic Errors Recovery (CER).
ENGR00152593	[LIN] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00141763	[LIN] Add tests for 2 IP instances on the same platform	Create a test which is exercising simultaneously two IPs (FlexLIN or ESCI) in the same time.
ENGR00140634	[LIN] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00152500	[LIN] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA

ENGR00140633	[LIN] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>
ENGR00141673	[LIN] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".

ENGR00138717	[LIN] Lin driver does not works properly when eSCI_C, eSCI_D, eSCI_E, eSCI_F, eSCI_G, eSCI	The variable Lin_ChannelHardwareMap[LIN_HW_MAX_MODULES] holds the value for the total no of lin channels configured in tresos and not for the total available lin channels. Plese see an example to illustrate this issue: Suppose two lin channels (eSCI_C and eSCI_D) are configured by user in tresos. The variable Lin_ChannelHardwareMap[LIN_HW_MAX_MODULES] in Lin.c file is used for mapping the hardware channels from the corresponding logical lin channels. So in this case the macro LIN_HW_MAX_MODULES will be equal to 2 (Lin_ChannelHardwareMap[0] and Lin_ChannelHardwareMap[1]) whereas the hardware channel numbers are 2 and 3 for eSCI_C and eSCI_D channels respectively which are outside the array boundary of variable "Lin_ChannelHardwareMap". Due to this the interrupt handler “ESCI_LLD_InterruptHandler()” for the corresponding eSCI_C and eSCI_D lin channels is not processed.
ENGR00156447	[LIN] Lin_GetStatus() is wrong implemented	NewWork Description: Lin_GetStatus() is wrong implemented. The Lin driver has to check only the pointer and has to setup the pointer to pointer value with an address, where the status info is located. In the current implementation it copy the receive data buffer to the location provided by LinSduPtr instead of providing only the address of internal DataBuffer A recommended implementation is: Lin_StatusType Lin_GetStatus(uint8 Channel, uint8 **LinSduPtr) { . . . . if(Chan_State == LIN_RX_OK) { /* Provide the address of data received */ *LinSduPtr = SduBuffAddress; } . . . return Chan_State; } Also for more details please refer to: - http://www.autosar.org/bugzilla/show_bug.cgi?id=17751 - http://www.autosar.org/bugzilla/show_bug.cgi?id=18826 These CR also invalidates the implementation of ENGR156021. Expected behavior: […]
ENGR00153805	[LIN] Lin_GetStatus() pointer check wrong	When the LinIf calls Lin_GetStatus() it supplies a pointer which should be used by Lin_GetStatus() to write its status to. Currently a pointer to pointer is given to the Lin Driver. The pointer is created in the LinIf_MainFunction and points to NULL. Lin_GetStatus() checks for a NULL pointer and returns NOK when the pointer Lin_SduPtr is null. The Driver has to check only the pointer and has to setup the pointer to pointer value with an address, where the status info is located. The Driver does not have to check the pointer to pointer value, because itself has to provide this value.

ENGR00154795	[LIN] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00141208	[LIN] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Attached to this CR are the templates for Excel and HTML. For Excel files, for the documents that have revision history (ex: Traceability Matrix, EPD test report), use attached "JDP header - with revision history.xlsx" The other 2 attachments will be used by the tools generating the other documents.
ENGR00155121	[LIN] Request to add LIN_ prefix to macro definitions in Lin.h file	Modification request on the LIN driver: In Lin.h header file there is a define: #define NO_ERROR 0x00U The name of this macro can cause problems when another module, which includes Lin.h, wants to use this name for an enum entry for example. As a specific example, the DCM module uses the name NO_ERROR as an enum element. Now, if any module includes both DCM and LIN header files, an error will be generated when compiled. Maybe the name should be changed into #define LIN_NO_ERROR 0x00U There are several defines in this situation (please check all of them).

ENGR00152231	[LIN] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).
ENGR00141757	[LIN] mismatch / missing XDM/XML header	Some modules XDM do not contain a header (e.g.gpt.xdm) or contain a header and a duplication of the copyright (e.g. can.xdm).
ENGR00142689	[LIN] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header The following duplicated copyright must be removed: <!-- ** Copyright 2006-2011 by Freescale Semiconductor Inc and STMicroelectronics ** All rights exclusively reserved for Freescale Semiconductor Inc and STMicroelectronics, ** unless expressly agreed to otherwise. -->
ENGR00157276	[MCU] Wrong DMA configuration	Problem detailed description (how to reproduce it): The EDMA_CR register is wrong configured by MCU driver when it initializes the DMA. (Call of DMA_Init APs). The GPRxPRI fields of CR register are cleared. Also these fields are not available at plug-in level. Due to all channel groups have the same priorities with Round-robin arbitration is disabled the DMA channels are "inactive" => All drivers which use DMA for data transfer (LIN/ADC/SPI) are not working Workaround: - the only workaround we have is to reconfigure the EDMA_CR register after the MCU_Init is called. REG_WRITE32(0xFFF44000, 0x0000E400 Preconditions: NA Trigger: NA Observed behavior: at runtime Expected behavior: NA Proposed solution (Optional): The MCU plugin need to be updated to support all bitfields of EDMA_CR register (check the spec.)
ENGR00140923	[MCU] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00152591	[MCU] Add EPD testing to Test Specification and implement with V&V	NewWork Description: [Add EPD testing to Test Specification and implement with V&V] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00140608	[MCU] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00142894	[MCU] BSWMD files are not supplied with the MCAL	Problem detailed description: The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. Observed behaviour: The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-PARAMETER-DEFINITION container. Problem is that this container, which according to AUTOSAR_InteroperabilityOfAuthoringTools.pdf and AUTOSAR_Methodology.pdf, is a non splittable container that cannot be divided between files.
ENGR00152499	[MCU] Compilation tests using TmakGen	NewWork Description: Use TmakGen script to generate the compilation tests. Expected behavior: All combination of compilation flags are generated and all compilation tests passed. Requirement source: Testing Strategy Proposed solution (Optional): NA

ENGR00136111	[MCU] Compile error because of incompatible initializer types	Package: Mcu MPC56xxL Found in version: MCAL_3.0_LEOPARD_BETA_HF5_0.9.0 What happens (symptoms): ---------- If more than 3 RAM Sector Configuration Sections are defined the last of these section is generated wrong. * MISRA-C:2004 19.1 VIOLATION: Refer to REF 1 above*/ {/* start of Mcu_RamConfig[3] */ (uint32)Gpt_Cfg_Ptr,/*section base address (must be aligned to 4 bytes) */ (uint32)0x00000004UL,/* section size in bytes (must be multiple of 4) */ (uint32)0x00000000ul /* value to be filled with */ }, /* end of Mcu_RamConfig[3] */ /* MISRA-C:2004 19.1 VIOLATION: Refer to REF 1 above*/ {/* start of Mcu_RamConfig[4] */ Gpt_Cfg_Ptr, /* section base address (must be aligned to 4 bytes) */ (uint32)0x00000004UL, /* section size in bytes (must be multiple of 4) */ (uint32)0x00000000ul /* value to be filled with */ } /* end of Mcu_RamConfig[4] */ Two RAM sections with the same configuration in configuration tool are generated in two different ways as can be seen above. Because of the missing cast following compile error occurs: "GenData/src/Mcu_Cfg.c", line 168: error (dcc:1552): initializer type `unsigned int' incompatible with object type `unsigned int *' ..\..\MakeSupport\cmd\make: *** [obj/Mcu_Cfg.o] Error 1
ENGR00139752	[MCU] Compiler warnings for Mcu.c and Mcu_Irq.c	Found: MPC560xB in MCAL3.0 RTM 3.0.1 The following compiler warnings are notified by Windriver DiabData 5.7.0.0: ~/Mcu/Mcu.c:685: warning: file does not end in newline ~/Mcu/Mcu_Irq.c:262: warning: file does not end in newline
ENGR00133824	[MCU] Configurable option to use / not use the MCAL MCU to enter Low Power Modes	Please provide a Configurable option to use / not use the MCAL MCU to enter Low Power Modes(STANDBY, STOP, HALT) as in MCAL AS2.1. As per CQT00003723: The implementation specific parameter "EnterLowPowerMode". Actual transition to Mcu low power mode will be done in Mcu_SetMode() function if the parameter "EnterLowPowerMode" is "TRUE", If the parameter "EnterLowPowerMode" is "FALSE" actual transition to Mcu low power mode will be done in a customer function. Please see the Notes section for a full discussion on the topic.

ENGR00160958	[MCU] Configuration variable naming in generated code	The issue is referred to MCAL 3.0 MPC564xA version 2.0.0 RTM + HF2. The problem is that this customer uses another tool to configure the rest of the basic Software. Specifically, they configure the EcuM module, which contains a configuration of the startup sequence. Basically the configuration of the EcuM uses the epc files from Tresos to find the name of the configuration sets. But our code doesn't use the name of the configuration set... In the generated code, the configuration variable name is Adc_ConfigPB_x (where x is the index number of the configuration) in the post build case for ADC module. The customer reports the problem for Adc and Wdg, I believe it is also true for FR. Apparently other modules use directly the name of the configuration set defined in the xdm/epc (maybe this needs to be verified for all modules), and the customer needs this behavior for all modules.
ENGR00140409	[MCU] Consistency check is not correct for some of the configuration Parameters	The following parameters do not generate an error when input wrong value are inserted (Monaco): : McuNumberOfMcuModes McuRamSectors McuClockReferencePointFrequency McuExternalBusDivisonFactor McuPllClockFrequency ServerTimeSlot AddressPipeliningControl WriteWaitStateControl ReadWaitStateControl McuLineBufferConfiguration McuRamSectionSize
ENGR00140623	[MCU] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>

ENGR00152617	[MCU] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00151562	[MCU] Exclusive areas information to be provided as part of the driver documentation	For each exclusive area (EA) please include in the IM the ASR API or ISRs names that call the LLD function containing the EA. If this path is affected by configuration, the dependency will be described (config parameters).
ENGR00141289	[MCU] Filename issue	The file Siu_Mcu_LLD.c is included as SIU_Mcu_LLD.h within Mcu_LLD.h, this may be an issue on an OS that distinguishes between lower cases and upper cases.
ENGR00139937	[MCU] Inconsistencies in design document	The AUTOSAR_MCAL_MCU_SDD.EAP does not contain the following : 1. Proper Design Id 2. File Structure Package 3. All the logical objects and containers in the driver Plugin 4. State Diagram 5. Activity diagram
ENGR00160578	[MCU] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'
ENGR00152949	[MCU] Internal refractoring for fixing misra violation in Mcu_SetMode api	NewWork Description: Mcu_LLD_Apply_Mode is returning a Std_ReturnType and it is not recieved to any parameter (not utilized) Which generatess MISRA 2004 Rule 16.10 violation Expected behavior: MISRA 2004 Rule 16.10 Violation should be fixed Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140244	[MCU] MCU user manual, incomplete list of return values	In 3.5.4 Function Mcu_GetResetReason, the list of return values of the API is incomplete (only 4 values are listed, although the parameter Mcu_ResetType can have much more values, as documented in 3.8.3 Enumeration Mcu_ResetType).

ENGR00153084	[MCU] MCU_START_SEC_RAMCODE is not explained in the MCU driver documentation	Problem detailed description (how to reproduce it): MCU_START_SEC_RAMCODE is not explained in the MCU driver documentation Can you please update the documentation and give a hint what this section is good for and what the integrator is supposed to to with this section in the Memmap.h. Preconditions: Documentation Trigger: NA Observed behavior: MCU_START_SEC_RAMCODE is not explained in the MCU driver documentation When can it be observed? (at configuration time, at runtime, at compile time?) Documentation Expected behavior: The chapter "Sections to be defined in MemMap.h" from the IM must include this info. Note: in the “Expected behavior” field, please mention also the requirement source too (cPRD, Errata version, RM, CR number etc.) Proposed solution (Optional): The chapter "Sections to be defined in MemMap.h" from the IM must include this info.
ENGR00152937	[MCU] Move to PCLint version 9.00F	NewWork Description: [Move to PCLint version 9.00F, review PCLint options, esp. adding the new supported rules 12.5 and 12.6] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140910	[MCU] Multiple Configuration not implemented	Plug-in generates only the first configuration when more than one configuration is created.
ENGR00140067	[MCU] Plugin generation error when PLL is disabled	Plugin generation error when PLL is disabled and crystal oscillator is selected as clock source.

ENGR00154794	[MCU] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00142880	[MCU] Request to check memory mapping implementation in all MCAL modules	Please review/check memory mapping in all MCAL modules to ensure all the data/code is placed in the correct sections as defined by Autosar spec. Customer requests this as recently there were found bugs in MCAL2.1 MPC5510 and MPC560xB (ENGR140612, ENGR00134104) where configuration data were not placed into correct sections in memory (i.e. memory sections incorrectly named, configuration arrays defined without const going into .data section instead of const section etc.).
ENGR00139894	[MCU] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format. Also the GPT test plan template is attached.

ENGR00156917	[MCU] Wrong MODULE_REF and ADMIN-DATA attributes in the plugin schema file	Problem detailed description (how to reproduce it): The following MODULE_REF reference is incorrect: <d:lst name="MODULE_REF"> <d:ref type="MODULE_REF" value="ASPath:/M4_XDM_AR_PKG_NAME/Mcu"/> </d:lst> It should be defined as: <d:lst name="MODULE_REF"> <d:ref type="MODULE_REF" value="ASPath:/M4_XDM_AR_PKG_NAME/M4_XDM_AR_MODULE_NAME"/> </d:lst> Also ADMIN-DATA should be updated as follow: <ad:ADMIN-DATA> <ad:LANGUAGE>EN</ad:LANGUAGE> <ad:DOC-REVISIONS> <ad:DOC-REVISION> <ad:REVISION- LABEL>M4_XDM_SW_VERSION_MAJOR.M4_XDM_SW_VERSION_MINOR.M4_XDM_SW_VERSION_PATCH</ad:RE VISION-LABEL> <ad:ISSUED-BY>M4_XDM_AR_MODULE_ORIGIN</ad:ISSUED-BY> <ad:MODIFICATIONS> <ad:MODIFICATION> <ad:CHANGE> <ad:L-2 L="EN">Release M4_XDM_SW_VERSION_MAJOR.M4_XDM_SW_VERSION_MINOR.M4_XDM_SW_VERSION_PATCH</ad:L-2> </ad:CHANGE> <ad:REASON> <ad:L-2 L="EN">Release M4_XDM_SW_VERSION_MAJOR.M4_XDM_SW_VERSION_MINOR.M4_XDM_SW_VERSION_PATCH</ad:L-2> </ad:REASON> </ad:MODIFICATION> </ad:MODIFICATIONS>
ENGR00141797	[MCU] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00151661	[MCU]: Remove unnecessary doxygen tags to remove doxygen warnings	NewWork Description: [doxygen is reporting warning for tags whose value is mentioned as "None"] Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [Quality]
ENGR00140924	[PORT] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00140604	[PORT] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.

ENGR00142899	[PORT] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-
ENGR00151402	[PORT] DET check for ‘Direction’ in API Port_SetPinDirection (Pin,Direction)	In the API Port_SetPinDirection (Pin,Direction) , the parameter ‘Direction’ was not checked when DET was ON. This check is required.
ENGR00137780	[PORT] Default value for PortPinPcr incorrect	The calculation of the default value (and such also the calculate button) goes wrong if the PortPins are located in different PortContainers using the same ID (for example two PortContainer containing one PortPin each will calculate PortPinPcr value of 83 for both of them).
ENGR00153658	[PORT] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>
ENGR00139929	[PORT] Inconsistencies in design document	The AUTOSAR_MCAL_PORT_SDD.EAP does not contain the following : 1. File Structure Package 2. Configuration Data Module 3. Sequence Diagram

ENGR00141676	[PORT] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00151408	[PORT] NULL pointer check in API Port_GetVersionInfo(Std_VersionInfoType * pVersionInfo) for	NULL pointer check in API Port_GetVersionInfo(Std_VersionInfoType * pVersionInfo) for the parameter 'pVersionInfo' required when DET is ON.
ENGR00139508	[PORT] PCR 10 and 11 ODE option not allowed	Customer has an issue when configuring the MCALs (version3.0.1 for the Bolero B family) for PCRs 10 and 11. They need to use an open drain configuration for these two pins in their application. The open drain output is a supported configuration for PCR10 and 11 and should be supported in the Tresos tool. The Tresos tool does not allow the ODE selection once the I2C alternate functions are selected. Here attached the screen shots of the configuration of these two pins with the ODE disabled. The customer has manually configured via modifying the MCAL output for these two PCR registers. The ODE function is available on the device but not in Tresos during the configuration.
ENGR00139187	[PORT] Port driver optimization	Is it necessary to generate the large arrays Port_PinDescription and Pad_funct_extrasettings (PSMI), if PORT_SET_PIN_MODE_API == STD_OFF? Currently these arrays are generated always. It seems, that only the functions Port_SetPinMode(pin,mode), PORT_Siul_SetPinMode(pin,mode) and Siul_PSMI_PadSet(pin,mode) use this arrays. These functions are disabled, with PORT_SET_PIN_MODE_API == STD_OFF. I checked this with MCAL 3.0.1 and get the same impression - these arrays are defined but not used if PORT_SET_PIN_MODE_API == STD_OFF. The compiler is not necessarily performing a dead stripping optimization hence this leads to a few hundred bytes being used for nothing.
ENGR00139895	[PORT] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format. Also the GPT test plan template is attached.

ENGR00140464	[PORT] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. --- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.
ENGR00141798	[PORT] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00141765	[PWM] Add tests for 2 or more IP instances on the same platform	In the case an IP has several instances on the same platform (e.g. 2 CAN controllers, or SPI ones), a test should be added to check that there are no interferences when both are running on the same app (see the CAN clone for an example of issue that went undetected until the customer found it).
ENGR00122673	[PWM] "catch all" issues for Adc, Pwm, Fls, Icu, see attachement	(source: Martin Markert talk w/ctmr) - Adc: Adc_Cfg.h: There is a validation rule missing that assures uniqueness of parameter "AdcResultBufferPointer" over all groups. - Adc: There is a validation rule missing that hw-triggered groups can not be used, when AdcHwTriggerApi is disabled. - Swt_LLD.c: Std_Types.h has to be included in WdgIf_Types.h, otherwise an compile error occurs. Vector's WdgIf_Types.h does not include this file. - There is a validation rule missing that assures that a certain Emios channel is only used by one BSW module (Either Gpt or Icu or ...) - There is a validation rule missing that assures that MdisBit is not set in Mcu when an Emios channel is configured. - Use a meaningful name for checkbox "MdisBit" in module Mcu (e.g. "Disable Emios Unit") - Pwm does not work with Pwm_Polarity_High and notification of type "FALLING_EDGE". In this case, a development error is reported. I could not find a documentation. Refer to eMIOS_Pwm_LLD.c Line 1432. - Fls does not work properly. The protection registers have to be set each time after reset. Normally, the registers should be non-volatile. - Icu integration manual: The names of the ISRs are not exactly the same as in the implementation. - Compiler abstraction for "PORT_FUNC" is not defined in Compiler_Cfg.h template
ENGR00140925	[PWM] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver

ENGR00123276	[PWM] - increase code coverage to 100%	increase code coverage to 100% by adding more tests
ENGR00136897	[PWM] A way must be provided to adjust the prescaler at mode transition	Request on All modules: In order to keep the peripheral input clock at same frequency in low power and high power a way must be provided to adjust the prescaler at mode transition. See implemented solution for MPC560x 2.1 with CQT00003735
ENGR00140847	[PWM] Add IOClient tests	Add IOClient functionality in PWM tests.
ENGR00140143	[PWM] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00142181	[PWM] Additional mapping of the logic channel names to the HW-channels	Context: In Pwm_Cfg.h, the logic channel names are mapped to channelIDs: #define PWM_CHANNEL0 0. In the Pwm_Cfg.c file, the HW-channels are used in the Pwm_ChannelConfigType structure: > EMIOS_0_CH_0 Problem: The mapping between logical and HW channels is calculated in PWM driver during runtime, which is inefficient; e.g. runtime of Pwm_SetDutycycle is approximately 2-4 µs. In comparison, a direct HW-register access, which would be possible with the above mentioned mapping, would take approximately 60ns. Suggestion: An additional mapping of the logic channel names to the HW-channels would help, i.e.: #define PWM_CHANNEL0_EMIOS_CH EMIOS_0_0 For improving the readability of the configuration files, this #define may be used in the Pwm_ChannelConfigType structure (Pwm_Cfg.c).
ENGR00142905	[PWM] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-

ENGR00140631	[PWM] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>
ENGR00122702	[PWM] Doc: Add exclusive area description in IM	PWM uses an exclusive area in the Flex_PWM, but its description is not included in the IM. IM should be corrected.
ENGR00151569	[PWM] Exclusive areas information to be provided as part of the driver documentation	For each exclusive area (EA) please include in the IM the ASR API or ISRs names that call the LLD function containing the EA. If this path is affected by configuration, the dependency will be described (config parameters).
ENGR00153723	[PWM] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>
ENGR00122910	[PWM] Include cPRD requirements in SRS	To update each driver SRS with cPRD requirements (see attached)
ENGR00139948	[PWM] Inconsistencies in design document	The AUTOSAR_MCAL_PWM_SDD.eap does not contain the following : 1. File Structure Package 2. enum values for each Attribute representing an ENUM 3. State machine diagram 4. All the logical objects and containers used in the driver Plugin

ENGR00141677	[PWM] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00136952	[PWM] Leopard RTM - Code Review	Please review the drivers against code review checklist (attached) and fix the findings
ENGR00137232	[PWM] MemMap Section PWM_START_SEC_CODE not closed	The section PWM_START_SEC_CODE is used in Pwm_LLD.h,but PWM_STOP_SEC_CODE is never called. PWM_START_SEC_CODE is then called a second time. Instead of this, PWM_STOP_CONFIG_DATA_UNSPECIFIED is called. See lines 135 + 160

ENGR00154799	[PWM] Proposal for MISRA warning syntax - ALL MCALs, ALL ARCHITECTURES	Problem detailed description (how to reproduce it): basically the customer would like to implement a MISRA warning explanation tool which reads the comments of our MCALsPreconditions. They propose some rules for comment and ask if we can do that. Below is customer's proposal. We are preparing an analysis tool for the MISRA warning explanation messages present in the MCAL code. For this, we would like to define some ground rules with you: - The mechanism applies only to MISRA 2004 justifications - References for Freescale justifications shall appear at the beginning of the file, and shall have the following format: /* REF <no_of_reference> - MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: <justification>*/ - A Freescale justification shall have the following format: /* MISRA-C:2004 <one or more MISRA rules separated with ","> VIOLATION: Refer to REF <no_of_reference> above */ Do you agree on this? Can you guarantee that this will not change in the future? [...] 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): [...]
ENGR00140151	[PWM] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Templates to be provided for Word, Excel an HTML.

ENGR00127243	[PWM] Replace tabulators with spaces	Tabulators indenting the code should be replaced by spaces. The list of files that mix tabs and spaces: ADCDig_LLD.c ADCDig_LLD.h ADCDig_LLD_CfgEx.h Adc.c Adc_Cfg.c Adc_Cfg.h Adc_Irq.c Adc_LLD.h Adc_PBcfg.c ESCI_LLD.c Fee.c Fr.c Gpt_LLD.c Lin.c Lin_Cfg.c Lin_Cfg.h Lin_Irq.c Lin_LLD.c Lin_LLD.h Lin_PBcfg.c Mcu.h Mcu_Cfg.c Mcu_LLD.h Mcu_PBcfg.c PWM_LLD.c Pwm.c Pwm_Cfg.c
ENGR00123973	[PWM] Review Checklists	Use the attached checklists to review the following: -design review -code review -req. traceability -SW reqs. -test spec.
ENGR00139899	[PWM] The quality documentation contains old forbidden Freescale logo	The documents like AUTOSAR_MCAL_ADC_TP.doc and similar contain Freescale logo with text "Launched by Motorola" within document footer. This logo is not allowed to be used since May 2005 ! In the attachment there is proper logo in .WMF format. Also the GPT test plan template is attached.
ENGR00131848	[PWM] UM/IM Automated Documentation	Please create the artifacts needed to create the UM/IM documentation by using autometic tool chain.
ENGR00121683	[PWM] Update Pwm Leopard UM	please update the Pwm Leopard UM with details regarding channel specific parameters.
ENGR00140693	[PWM] Update user and integration manuals	For PWM UM/IM docs implement documentation review report.

ENGR00139123	[PWM] Use driver data types instead of generic ones where applicable	Other modules are probably affected as well (beside SPI). Affected parts should be identified by this inquiry. ---------- This is only a formal thing. The Spi API function declarations use their own data types for the handles. E.g. here (Spi.h): FUNC (Std_ReturnType, SPI_CODE) Spi_AsyncTransmit(VAR(Spi_SequenceType, AUTOMATIC) Sequence); However, the generated handles as generated into Spi_Cfg.h use generic data types. E.g. here: [!LOOP "SpiDriver/*/SpiChannel/*"!][!// #define [!"name(.)"!] (uint8)[!"@index"!]u [!ENDLOOP!][!// No compile warning or error is generated since (Spi_SequenceType) is already defined to (uint8). However, it would be "nicer" to have the handles directly casted to their destined data type.
ENGR00124037	[PWM] VSMD check on epd files	Check the epd files against VSMD
ENGR00131438	[PWM] When a STS register clears, 0x3FFF is written	By module "FlexPwm_Pwm_LLD_Init" and "FlexPwm_Pwm_LLD_Init", when a STS register clears, 0x3FFF is written. But no 11-8bits of a STS register are assigned. Why does it write 0x3FFF? REG_WRITE16(FlexPWM_SubRegs[SubModuleId+SubModuleOffset]+FLEXPWM_STS, (uint16)0x3FFF); Corresponding section: ~ \MPC56xxL_MCAL_3.0_LEOPARD_BETA_HF5_0.9.1_signed \Pwm_TS_T2D17M0I9R0\FlexPWM_Pwm_LLD.c(467): FlexPwm_Pwm_LLD_Init The same access processing as another places exists. - FlexPWM_Pwm_LLD.c(679): FlexPwm_Pwm_LLD_DeInit
ENGR00131880	[PWM] Zero division correction	Please correct the following zero modulo division or document the limitation in the UM/IM: Bolero CS ---------- element * HF_MCAL_3.0_BOLERO_RTM_HF3_3.0.0 element * HF_MCAL_3.0_BOLERO_RTM_HF2_3.0.0 element * HF_MCAL_3.0_BOLERO_RTM_HF1_3.0.0 element * BLN_MCAL_3.0_BOLERO_RTM_3.0.0 " File: ..\AUTOSAR\pwm\specific\src\eMIOS_PWM_LLD.c tempDuty32 = ((uint32)tempDuty + PwmEmiosSpecificParam->Pwm_Offset) % Period ; (Line No. 917, 1443) """Period"" value is depending the unified channel registers EMIOS_CADR, EMIOS_CBDR registers. When the counter bus is started the registers EMIOS_CADR, EMIOS_CBDR are initialized with non-zero values. So, care should be taken such that the counter bus should be started before initializing a channel in OPWMB / OPWMT mode. If counter bus is not started before initializing a channel in OPWMB / OPWMT mode, then care should be taken such that ""Period"" should not be zero in the driver code as mentioned in condition column. This can be documented in UM or IM."
ENGR00120093	[PWM] define fault functionality features	determine needed features for the fault functionality and update the PWM driver accordingly. Add tests for the implemented features. This feature is only for platforms based on eTimer and FlexPWM peripherals. PR-MCAL-3162 and PR-MCAL-3163
ENGR00141802	[PWM] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header

ENGR00123961	[PWM] xxxID not used to generate symbolic name	MCAL MPC5510 Some MCAL modules do not used the xxxID parameter from the configuration. This could be a source for incoherence between modules that have indirect references to the MCAL module in the configuration. The following MCAL modules have been identified: - ADC: AdcHwUnitId parameter - MCU: McuMode parameter - SPI: SpiChannelId, SpiJobId, SpiSequenceId - WDG: WdgIndex parameter These parameters are not used by their modules generation script. However, since they are defined by the AUTOSAR specification, it would mean that a symbolic name should be give a value defined by the xxxID parameter. In this way, all modules that are using an integer value to reference the symbolic name know what to expect.
ENGR00162383	[RESOURCE] Inconsistency in xdm file for UUID values	Problem detailed description (how to reproduce it): The prefix “ECUC” is missing for the UUID value’s for several parameters. Preconditions: [...] Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) [...] Expected behavior: [...] Proposed solution (Optional): Implementation should be as below "<a:a name="UUID" value="ECUC:72d846f3-bcd3-422c-97e2-8ea89ec3f2ed"/>" as per 'AUTOSAR_EcucParamDef.arxml'
ENGR00141804	[RESOURCE] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00138693	[SPI] - Array index bounds must be checked before referencing array values	Spi_SyncTransmit() initializes local variables before checking the bound limits for Sequence parameter P2VAR(Spi_SequenceState, AUTOMATIC, SPI_VAR) pSequenceState = &SpiSequenceState[Sequence]; VAR(uint32, AUTOMATIC) SpiSequenceUsedHWUnits = SpiSeqUsedHWUnits[Sequence];
ENGR00139786	[SPI] - Code traceability must be checked against Design IDs	According to the process, the code traceability must be checked against design IDs, using "@remarks Implements DSPIxxxx" directives; only the situations when design IDs are not applicable must be traced against Requirement IDs At this moment, the SPI driver traceability is checked against Requirement IDs in almost all cases.
ENGR00151589	[SPI] - Fix the review findings as in peer review 38859461	NewWork Description: Below atre the review fidnings to be fixed in Spi plugin files: • “SpistartJobNotification” – the function name is not consistent with the rest of the function notations. • “Spistartjobnotificationenable” – the variable’s name in Spi.xdm is not consistent with the rest of the notations. • The description of “Spistartjobnotificationenable” in Spi.xdm contains a typo: “notifictaion”. • In “Spi_StartNotifications.c” file from “test_spi/generic/src”, the “@file” tag remained “Spi_EndNotifications.c” instead of “Spi_StartNotifications.c” Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]

ENGR00138680	[SPI] - Improve SC/BC from 95% to 100%	9 test cases needed for SC/BC improvement
ENGR00139031	[SPI] - MISRA violation on E_OK & E_NOT_OK	Attributions and comparisons between Std_ReturnType variables (uint8) and E_OK & E_NOT_OK values (signed integer) are violating MISRA, due to the type mismatch.
ENGR00140114	[SPI] - Spi_GetVersionInfo() - parameter VersionInfo must be checked	Into the function Spi_GetVersionInfo() implementation: 1) VersionInfo pointer must be checked against NULL value. 2) The function shall be available even if Spi_Init() was not called for the driver => the check on SpiConfigPtr shall not be made for Spi_GetVersionInfo()
ENGR00140926	[SPI] - correct wrong design ids from code/design document	There are several design IDs which only appear in the source file (code) but do not appear in the design document (.eap). These IDs must be corrected in order to have a good traceability report. They either must be added in the design document or corrected in the source code. See attached files for the list of errors for each driver
ENGR00140140	[SPI] Add tests for Multiple Configurations	Create a test that checks the multiple configurations feature of the driver. It means that 2 or more configurations will be used as part of the same test case.
ENGR00139790	[SPI] Allow more than 256 SPI sequences	Customer request: We have 5 SPI driven external devices, each using around 60 SPI Sequences, so we really need to configure more than 255 sequences. CE note: There is no benefit from restricting this type to uint8. As such we can envision a non-Autosar feature to allow more than 256 sequences. To keep standard conformance on the default case, we will keep this restriction by default. We will add a checkbox that, if checked, will remove the restriction from the configuration, allowing (2^32)-1 sequences. This will apply not just to sequences, but also to channels and jobs numbers.
ENGR00141471	[SPI] Allow the same short name for symbolic names of objs with the same ID in multiple configs	If the same SPI channel is included in several post build configurations, it should be allowed to have the same short name (the driver to generate the same symbolic name) as long as the ID is identical accross configurations. Also, generate the symbolic name define (#define Name ID) only once for such objects (same name and ID) included in multiple configurations. Rationale: This will allow the upper layer code to be the same when using different post build configurations. See also the NoteLog for an example and a possible side-effect if this is not implemented.
ENGR00142906	[SPI] BSWMD files are not supplied with the MCAL	The BSWMD files as a part of the autosar concept as described in the AUTOSAR_BSWMDTemplate.pdf and are necessary to complete the integration. The SchM module is depending on information about for example which exclusive areas, scheduled entities, event, startup and shutdown functionality is imported from the BSWMD file. It is necessary to supply VendorID and APIInfix in the BSWMD file for modules where it is possible that several different instances of a module can be present in the configuration. The reason for this is for example different kinds of CAN controllers in a micro controller or that an external CAN controller is attached to supplement the controllers provided in the micro One example of this is the CAN driver module where a call to the function Can_Write would look like: FUNC (Can_ReturnType, CAN_CODE) Can_43_A_Write(...); Since it it possible to combine the information from several concepts, we combine the VSMD (Vendor Specific Module Definition AUTOSAR_ECU_Configuration.pdf) and BSWMD information in one file, we have added the BSWMD information to the epd files to be able to generate the final integration configuration, the CAN file is attached for reference. The BSWMD files are supposed to carry information about which modules it references. This is done in the ECU-

ENGR00142161	[SPI] COSMIC compiler integration	The COSMIC compiler should be used for This activity need some modifications in the code because the actual implementation is not supported by the COSMIC compiler:: 1.Some INLINE functions declared after their usage, should be moved in the upper part of the following files: a.Spi.c b.Spi_LLD.c 2.The SchM files contain the following functions that should be replaced as follow: ASM_KEYWORD uint32 Adc_schm_read_msr(void) { mfmsr r3 } Should be replaced as: uint32 Adc_schm_read_msr(void) { ASM_KEYWORD("mfmsr r3"); }
ENGR00140627	[SPI] Copyright year must be updated to 2011	Copyright year must be updated to 2011 in UM / IM xml templates for each driver. The XML template used for generating the documentation for each MCAL drivers (UM / IM) contains a variable used by the Docato Publication System in the the last page of the manual: "Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc." The issue comes from the following lines: <docmeta> <document-id> <document-partnum>IM17ICUASR3.0R1.0.0</document-partnum> </document-id> <document-rights> <copyrfirst> <year>2010</year> </copyrfirst> <copyrlast> <year/> </copyrlast> <doc-owner> <organization>Freescale Semiconductor, Inc.</organization> </doc-owner> </document-rights> </docmeta>

ENGR00152632	[SPI] Detailed description of DEM events	We had today a brider internal discussion on this topic, i.e. on how to handle errors reported by MCAL to DEM. The conclusion was, that it is unclear for us, which of the DEM Errors are fatal errors, i.e. that indicate that ECU is defective and needs to be replaced. Therefore, we need from your side additional information for each DEM error, that can be set by Adc, Fls, Mcu and Pwm driver (including affected Low Level Drivers): 1) under which conditions is such a DEM Error set ? 2) which error is indicating a fatal HW error, i.e. non recoverable error ? >> can you provide us such a description ? Notes: 1) this issue is quite urgent, as it might affect the start of series production for one of our customers 2) we already studied AUTOSAR description of these DEM Errors, but that description is way too general 3) we do not ask how to process these errors, we see this as our responsibility 4) we just need to understand the significance of these DEM errors
ENGR00140435	[SPI] Documentation update	Update the documentation as described in the analysis.
ENGR00141625	[SPI] Documentation update - remaining actions	Please see the attach file.
ENGR00159687	[SPI] Driver generates compiler warnings when compiled with DIAB 5.8.0.0	Problem detailed description (how to reproduce it): While compiling with DIAB 5.8.0.0, SPI driver generates below compiler warnings. "narrowing or signed-to-unsigned type conversion found: " Preconditions: DIAB 5.8.0.0 compiler version shall be used for building driver. Trigger: [...] Observed behavior: Compiler warning is generated during building driver. When can it be observed? (at configuration time, at runtime, at compile time?) compile time 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 the compiler warning in driver files. If it can not be fixed, update the comments in driver code with proper doxygen tags.
ENGR00151425	[SPI] Extended support for Micro Second Bus	NewWork Description: - SpiBaudRate Maximal baudrate for Spi in MSC mode was changed from 16MHz to the 40MHz - SpiTimeCs2Cs Definition of timing between chip select assertions was changed from microSec to the number of Tsck in TSB mode. Exact time between chip selects is then shown in tooltip of parameter in tresos configuration. Requirement source: - Customer Request See attached file for detailed description, and attached SPI.zip file for the modification added to the plugin.

ENGR00158596	[SPI] Fix MISRA violation	NewWork Description: MISRA violation 8.8 need to be fixed for SPI driver. object/function previously declared: 'Spi_JobTransferFinished' at location line 316, file C:\EB\tresos_2010_sr4\plugins\Spi_TS_T2D11M3I0R0\src\Dspi_LLD.c needs to be fixed Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): Update the SPI driver files to fix MISRA violation 8.8.
ENGR00153724	[SPI] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>
ENGR00141678	[SPI] Inconsistent list of files described in the drivers IMs	The list of files discribed in the drivers IMs is not consistent with the actual list of files delivered in the plugins: MCU module: Files required for compilation : Dma_types.h ---> doesn't exist Dmamux_types.h ---> doesn't exist in 3.0.1, only in 3.0.0. This file is not needed in MCAL 3.0.1 PWM module : eMIOS_Pwm_LLD.c ---> file required for compilation, but not in IM eMIOS_Pwm_LLD_IRQ.c ---> file required for compilation but not in IM SPI module : Dspi_LLD_CfgEx.h ---> does not appear in IM in section "Files required for compilation", but is present in the 3.0.1 package Dma_Spi_LLD.c ---> instead of Spi_Dma_LLD.c (in SPI IM) WDG module : Swt_LLD.h ---> not present in IM Wdg_LLD.h ---> appears twice in WDG IM LIN module : no source file(.c) is present in "Files required for compilation".
ENGR00136955	[SPI] Leopard RTM - Code Review	Please review the drivers against code review checklist (attached) and fix the findings
ENGR00140150	[SPI] Quality documents must contain FSL or JDP header	All Quality documents must contain the same header which includes title, company name(s), copyright, date, project, version, disclaimer, potentially a revision history, legend or other explanations. Templates to be provided for Word, Excel an HTML.

ENGR00139180	[SPI] SPI and DMA configuration question	The problem is that other SpiPhyTxDmaChannel SpiPhyTxDmaChannelAux SpiPhyRxDmaChannel configurations don’t work, but I cannot tell what is going wrong. Only saw that the default values worked and if they were changed the Spi communication fails.
ENGR00138325	[SPI] SPI shall contain an empty string for VendorApiInfix	This FR report was stated to affect also SPI, hence it was cloned for it (see description below for FR) Since AUTOSAR FrIf supports usage of multiple Fr modules, the Fr modules might use unique API names according to the following rule: Fr_<VendorId>_<VendorApiInfix>_FunctionName() The parameters VendorId and VendorApiInfix are derived from the CommonPublishedInformation container of the Fr module. Thus for proper operation between Fr and FrIf, the configuration parameters VendorId and VendorApiInfix must be set properly. For drivers which do not use the Infix, providing standard API names (e.g. Fr_FunctionName()), the VendorApiInfix must be empty (empty string). The setting of the default values for those configuration parameters is essential, since those parameters are not editable by the user and newly created configurations derive the default values. For Leopard MCAL 0.9.2 it was observed that the VendorApiInfix of the Fr is "0" which finally leads to a linkage problem between Fr and FrIf. The current implementation violates BSW00347 of the AUTOSAR document "General Requirements on Basic Software Modules" See BSW00310 and BSW00347 of the AUTOSAR Document "General Requirements on Basic Software Modules" and the
ENGR00143264	[SPI] Shared channel validation shall not be applied on invalid channel fields	When checking if channels in different PB SPI configurations can share the same channel name, only valid channel fileds shall be checked for having the same values: - for an EB channel - SpiIbNBuffers shall not be checked - for an IB channel - SpiEbMaxLength shall not be checked impacted files: generate\Spi_PBcfg.c generate\Spi_Cfg.h
ENGR00159657	[SPI] Spi_DataType shall be defined at generation time to save RAM usage	NewWork Description: Spi_DataType shall be defined at generation time to save RAM usage. this is a feature request coming from one of our customer: currently, the Spi_DataType is statically defined to uint16. However, as customer only uses 8bit buffers for SPI, they would waste RAM by allocating 16bit. Thus, they request to determine at generation time the maximum size of bits needed for that data type and use the smallest possible c data type. Expected behavior: Requirement source: Customer Request Proposed solution (Optional):

ENGR00140242	[SPI] Spi_StartJobNotification needed	An Spi_StartJobNotification is not provided by SPI driver (not required in AUTOSAR). This issue has been discussed with FSL some time ago and it was rejected (ENGR00120339). However, the need for this API reappeared in the customer project, as solution to the following use case: Environment: - Bolero 04 (No DMA) - 2 devices sharing the same SPI bus - because of hard requirements to interrupt disabling only asynchronous SPi may be used - CS of one of these devices on non-CS pin Idea: - set CS in start/end-job notifications Additional req to SPI handler-Driver: StartJobNotification Old solution: - other controller - proprietary SW to handle SPI communication
ENGR00161916	[SPI] Spi_TP_019.c needs to be generic to test for all TSB supported platforms	NewWork Description: Spi_TP_019.c is designed to test TSB feature specifically for Monaco. Test pattern needs to modified to test for all platforms that is generic. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00140731	[SPI] Update tests for V&V	Update local tests to work on V&V.
ENGR00141472	[SPI] Update the SPI IM to describe how to use the same channel in multiple post build configs	See the discussion in ENGR131471 (including NoteLog)- there is a need to describe (based on the implemented driver checks) how to configure a channel in several post build configurations. Such an example needs to be included in the SPI IM as the current implemented plugin configuration checks are not obvious.
ENGR00141806	[SPI] remove duplicated copyright header from XDM files	Some modules XDM contain a duplication of the copyright (e.g. can.xdm) in the XDM file header
ENGR00131844	[SPI] review the traceability code vs req. vs test	There are several testable reqs. which have designID for which we created tests but the source files are not referenced in traceability matrix I.e.: ReqID DesignID File Ref Test SPI059DSPI02200(blank) Spi_TP004_TC001 Spi_TP004_TC006 Spi_TP004_TC002 Spi_TP004_TC003 Spi_TP004_TC005 Spi_TP006_TC001 Spi_TP004_TC007 DSPI02210(blank)Spi_TP004_TC001 Spi_TP004_TC006 Spi_TP004_TC002 Spi_TP004_TC003 Spi_TP004_TC005 Spi_TP006_TC001 Spi_TP004_TC007
ENGR00152374	[SPI] specific header file "mpc563m.h" should not be included in generic file (spi_tp_019.c)	In SPI all the test files are generic, specific header file is included in spi_tp_019.c, that need to be removed and to be added in spi_envseup.h

ENGR00153408	[SPI]New test cases needs to be added which are marked as testable & verification criteria is as	NewWork Description: The test cases which were marked as testable & verification criteria is as review needs to be implemented This type of requirements can be tested like this (example from CAN driver): #if defined(CAN_VERSION_INFO_API) #if (CAN_VERSION_INFO_API == STD_ON) EU_ASSERT(0U == 0U); #elif (CAN_VERSION_INFO_API == STD_OFF) EU_ASSERT(0U == 0U); #else EU_ASSERT(0U == 1U); #endif #else EU_ASSERT(0U == 1U); #endif Requirement source: Please refer Attached documents for further information (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00161458	[SPI]To fix VSMD error	NewWork Description: When the CR ENGR151425 is integrated, the following VSMD error is getting The node /TS_T2D22M9I1R0/Spi/SpiDriver/SpiExternalDevice/SpiBaudrate needs a MIN and MAX attribute. This is mandatory in VSMD. The node /TS_T2D22M9I1R0/Spi/SpiDriver/SpiExternalDevice/SpiTimeCs2Cs needs a MIN and MAX attribute. This is mandatory in VSMD. Expected behavior: […] Requirement source: […] (e.g. cPRD, gMRD, Customer Request, Quality, ASR SWS, RM.pdf, Errata.pdf...) Proposed solution (Optional): [...]
ENGR00139791	[WDG] The SWT_WDG_SOFT_LOCK and SWT_WDG_NO_SOFT_LOCK defines are not generated correctly	The SWT_WDG_SOFT_LOCK and SWT_WDG_NO_SOFT_LOCK defines are not generated correctly when the Wdg Operation Mode is MODE FAST.

ENGR00153659	[WDG] Handling correctly upper multiplicity in the XDM files	Problem detailed description (how to reproduce it): Upper multiplicity definition is not valid according to AUTOSAR standard parameter definition. Rule A202 is reported as violated by Vector Informatik GmbH - Amdc OEM Version 1.0.3.0 tool. Preconditions: Upper multiplicity definition. Trigger: [...] Observed behavior: [...] When can it be observed? (at configuration time, at runtime, at compile time?) Configuration time Expected behavior: [...] Note: in the “Expected behavior” field, please mention also the requirement source too AUTOsar Proposed solution (Optional): UPPER-MULTIPLICITY definition in the XDM files has to be replaced from "*" to "1" <LOWER-MULTIPLICITY>1</LOWER-MULTIPLICITY> <UPPER-MULTIPLICITY>1</UPPER-MULTIPLICITY> <MULTIPLE-CONFIGURATION-CONTAINER>true</MULTIPLE-CONFIGURATION-CONTAINER>