MCUXpresso SDK Release Notes Supporting frdmkv31f

Change Logs

NXP Semiconductors

Contents

Driver Change Log

CLOCK 1

DSPI_CMSIS 2

I2C 2

UART 3

ADC16 3

CMP 3

COMMON 4

CRC 6

DAC 6

DMAMUX 6

DSPI 7

EDMA 9

EWM 11

FLASH 12

FTM 13

GPIO 15

I2C 16

LLWU 18

LPTMR 19

LPUART 19

Title	Page No.

LPUART_EDMA 22

LPUART_FREERTOS 22

PDB 23

PIT 23

PMC 24

PORT 24

RCM 25

SIM 25

SMC 25

UART 26

UART_EDMA 28

UART_FREERTOS 29

VREF 29

WDOG 29

Middleware Change Log

FatFs for MCUXpresso SDK 30

FreeMASTER Communication Driver 30

MOTOR_CONTROL for KSDK 31

RTCESL for KSDK 31

SAFETY_IEC60730B for KSDK 32

SD(SPI) Card driver for MCUXpresso SDK 32

Component Change Log

SERIAL_MANAGER 33

1 Driver Change Log

CLOCK

Current CLOCK driver version is 2.5.2

- 2.5.2
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 18.1
- 2.5.1
  - Bug Fixes
    - Fixed MISRA C-2012 issues, including rule 15.5, rule 16.4, rule 14.4, rule 10.1 and so on.
    - Fixed IAR warning Pa082 for the clock driver.
- 2.5.0
  - New feature:
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.4.0
  - New feature -Added two APIs to set slow and fast internal reference clock variable.
- 2.3.0
  - Fix the issue for MISRA-2012 check.
    - Fixed rule 10.4, rule 10.1, rule 10.6, rule 13.5, rule 10.8.
  - Bug Fix:
    - Fix incorrect External Oscillator Configuration sequence and ensure oscillator configuration be executed before it be enabled.
  - New feature
    - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.2.1
  - Bug Fix:
    - Fix the issue that MCG could not switch to FEE/FBE/PBE modes when OSCERCLK clock not enabled.
- 2.2.0
  - New Features:
    - [KPSDK-9157] Update CLOCK_SetFeiMode/CLOCK_SetFbiMode/CLOCK_BootToFeiMode() to support set MCG_C4[DMX32]=1 in FEI/FBI modes.
  - Bug Fix:
    - Update IP_CLOCKS array, remove unused gates and add missing gates.
- 2.1.0
  - Other Changes:
    - Merge fsl_mcg and fsl_osc into fsl_clock.
- 2.0.0
  - Initial version.

DSPI_CMSIS

Current dspi_cmsis driver version is 2.3

- 2.3
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.2
  - Bug Fixes
    - Fixed the bug that, the parameter num of APIs ARM_SPI_Transfer, ARM_SPI_Send and ARM_SPI_Receive, and the return value of API ARM_SPI_GetDataCount should be the number of data item defined by datawidth, rather than the number of byte.
- 2.1
  - Bug Fixes
    - Fixed the wrong clock polarity assignment in driver. For ARM_SPI_CPOL0_CPHA0 and other frame format parameters, CPOL = 0 means kSPI_ClockPolarityActiveHigh not kSPI_ClockPolarityActiveLow in driver.
  - New Features
    - Allowed user to set up the default transmit value by using ARM_SPI_SET_DEFAULT_-TX_VALUE.
    - Enabled slave select mode. Note this has no effect when user sets any of them because the driver can only support the hardware control function.
- 2.0
  - Initial version.

I2C

Current I2C CMSIS driver version is 2.2.0

- 2.2.0
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.0.1
  - Bug Fixes
    - In ARM_I2C_ABORT_TRANSFER operation in I2C_InterruptControl, the method to check if I2C is operating as slave is not correct, then master may have potential risk to block at the slave check code.
- 2.0.0
  - Initial version.

UART

The current UART CMSIS driver version is 2.1

- 2.1
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.48.610.110.310.411.111.914.415.716.116.316.416.617.720.720.9.
- 2.0
  - Initial version.

ADC16

The current ADC16 driver version is 2.3.0.

- 2.3.0
  - Improvements
    - Added new API ADC16_EnableAsynchronousClockOutput() to enable/disable ADACK output.
    - In ADC16_GetDefaultConfig(), set enableAsynchronousClock to false.
- 2.2.0
  - Improvements
    - Added hardware average mode in adc_config_t structure, then the hardware average mode can be set by invoking ADC16_Init() function.
- 2.1.0
  - New Features:
    - Supported KM series' new ADC reference voltage source, bandgap from PMC.
- 2.0.3
  - Bug Fixes
    - Fixed IAR warning Pa082: the order of volatile access should be defined.
- 2.0.2
  - Improvements
    - Used conversion control feature macro instead of that in IO map.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA-2012 rules.
      - Rule 16.4, 10.1, 13.2, 14.4 and 17.7.
- 2.0.0
  - Initial version

CMP

The current CMP driver version is 2.0.2.

- 2.0.2

    - Bug Fixes
        - Fixed the violations of MISRA 2012 rules:
            - Rule 10.3
- 2.0.1
    - Bug Fixes
        - Fixed MISRA-2012 rules.
            - Rule 14.4, rule 10.3, rule 10.1, rule 10.4 and rule 17.7.
- 2.0.0
    - Initial version.

COMMON

The current COMMON driver version is 2.3.2.

- 2.3.2
    - Improvements
        - Make driver aarch64 compatible
- 2.3.1
    - Bug Fixes
        - Fixed MAKE_VERSION overflow on 16-bit platforms.
- 2.3.0
    - Improvements
        - Split the driver to common part and CPU architecture related part.
- 2.2.10
    - Bug Fixes
        - Fixed the ATOMIC macros build error in cpp files.
- 2.2.9
    - Bug Fixes
        - Fixed MISRA C-2012 issue, 5.6, 5.8, 8.4, 8.5, 8.6, 10.1, 10.4, 17.7, 21.3.
        - Fixed SDK_Malloc issue that not allocate memory with required size.
- 2.2.8
    - Improvements
        - Included stddef.h header file for MDK tool chain.
    - New Features:
        - Added atomic modification macros.
- 2.2.7
    - Other Change
        - Added MECC status group definition.
- 2.2.6
    - Other Change
        - Added more status group definition.
    - Bug Fixes
        - Undef __VECTOR_TABLE to avoid duplicate definition in cmsis_clang.h
- 2.2.5

- Bug Fixes
    - Fixed MISRA C-2012 rule-15.5.
- 2.2.4
  - Bug Fixes
    - Fixed MISRA C-2012 rule-10.4.
- 2.2.3
  - New Features
    - Provided better accuracy of SDK_DelayAtLeastUs with DWT, use macro SDK_DELA-Y_USE_DWT to enable this feature.
    - Modified the Cortex-M7 delay count divisor based on latest tests on RT series boards, this setting lets result be closer to actual delay time.
- 2.2.2
  - New Features
    - Added include RTE_Components.h for CMSIS pack RTE.
- 2.2.1
  - Bug Fixes
    - Fixed violation of MISRA C-2012 Rule 3.1, 10.1, 10.3, 10.4, 11.6, 11.9.
- 2.2.0
  - New Features
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.1.4
  - New Features
    - Added OTFAD into status group.
- 2.1.3
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.3.
- 2.1.2
  - Improvements
    - Add SUPPRESS_FALL_THROUGH_WARNING() macro for the usage of suppressing fallthrough warning.
- 2.1.1
  - Bug Fixes
    - Deleted and optimized repeated macro.
- 2.1.0
  - New Features
    - Added IRQ operation for XCC toolchain.
    - Added group IDs for newly supported drivers.
- 2.0.2
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.4.
- 2.0.1
  - Improvements
    - Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.

    - Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTION" for specific SoCs which have no noncacheable sections, that helps avoid an unnecessary complex in link file and the startup file.
    - Updated the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
  - Initial version.

CRC

The current CRC driver version is 2.0.3.

- 2.0.3
  - Bug fix:
    - Fix MISRA issues.
- 2.0.2
  - Bug fix:
    - Fix MISRA issues.
- 2.0.1
  - Bug fix:
    - DATA and DATALL macro definition moved from header file to source file.
- 2.0.0
  - Initial version.

DAC

The current DAC driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed MISRA-2012 issues:
      - Rule 10.3, 10.8 and 17.7.
- 2.0.1
  - Bug Fixes
    - Moved the default DAC_Enable(..., true) from DAC_Init() to the application code so that users can enable the DAC's output.
- 2.0.0
  - Initial version.

DMAMUX

The current DMAMUX driver version is 2.0.5.

- 2.0.5
  - Improvements

    * Added feature FSL_FEATURE_DMAMUX_CHCFG_REGISTER_WIDTH for the difference of CHCFG register width.
- 2.0.4
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.4.
- 2.0.3
  - Bug Fixes
    * Fixed the issue for MISRA-2012 check.
      · Fixed rule 10.4 and rule 10.3.
- 2.0.2
  - New Features
    * Added an always-on enable feature to a DMA channel for ULP1 DMAMUX support.
- 2.0.1
  - Bug Fixes
    * Fixed the build warning issue by changing the type of parameter source from uint8_t to uint32_t when setting DMA request source in DMAMUX_SetSourceChange.
- 2.0.0
  - Initial version.

DSPI

The current dspi driver version is 2.2.4.

- 2.2.4
  - Bug Fixes
    * Fixed bug that instance with shared TR/RX EDMA request cannot transfer 1 datawidth of data as master in single transfer.
- 2.2.3
  - Improvements
    * Added macro of getting maximum transfer size using EDMA way.
- 2.2.2
  - Bug Fixes
    * MISRA C-2012 issue fixed:
      · Fixed rules, containing: 10.8, 10.3.
    * Fixed the build warning issue.
    * Fixed the bug that PCS would temporarily pull down for a while during master initialization.
    * Fixed compiling error of undefined identifier g_dspiDummyData.
    * Changed the type of kDSPI_AllStatusFlag and kDSPI_AllInterruptEnable from int to uint32_t to fix Pe068 warning of integer conversion.
    * Eliminated IAR Pa082 warning in DSPI_MasterTransferDMA.
- 2.2.1
  - Bug Fixes
    * Fixed the bug for double execution of transfer complete callback in master interrupt

transfer mode. In the interrupt routine, the DSPI interrupt may drop in the situation of the interrupt pending by itself while receiving the last frame, adding check to the transfer state to execute the callback function.
    * Fixed wrong logic in DSPI_SetFifoEnable().
    * MISRA C-2012 issue fixed.
      · Fixed rules, containing: rule-12.1, rule-17.7, rule-16.4, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.1, rule-10.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3, and rule-8.5.
- 2.2.0
  - New Features
    * Added gasket feature for SPI EDMA driver, which reduces one channel used in the EDMA master transfer. With this feature support, only two channels are needed. For example, if the gasket feature is supported, we could use the DSPI_MasterTransferCreateHandleEDMA function like below: DSPI_MasterTransferCreateHandleEDMA(EXAMPLE_DSPI_MASTER_BASEADDR, &g_dspi_edma_m_handle, DSPI_MasterUserCallback, &userData, &dspiEdmaMasterRxRegToRxDataHandle, NULL, &dspiEdmaMasterIntermediaryToTxRegHandle);
    * Added dummy data setup API to allow users to configure the dummy data to be transferred.
    * Added new APIs for half-duplex transfer function. Users can send and receive data by one API in the polling/interrupt/EDMA way, and they can choose either to transmit first or to receive first. Additionally, the PCS pin can be configured as assert status in transmission (between transmit and receive) by setting the isPcsAssertInTransfer to true.
- 2.1.4
  - Bug Fixes
    * DSPI EDMA driver: the DSPI instance has been separated, so the DMA request source can now transfer up to 32767 Bytes data in one DSPI_MasterTransferEDMA() transfer.
- 2.1.3
  - Bug Fixes
    * DSPI EDMA driver can no longer support the case that the transfer data size is odd, but the bitsPerFrame is greater than 8.
  - Improvements
    * Added #ifndef/#endif to allow users to change the default TX value at compile time.
- 2.1.2
  - Bug Fixes
    * DSPI_MasterTransferBlocking function would hang in some corner cases (for example, some cases with bitsPerFrame is 4,6 and kDSPI_MasterPcsContinuous transfer mode).
- 2.1.1
  - Bug Fixes
    * Set the EOQ (End Of Queue) bit to TRUE for the last transfer in transactional APIs.
- 2.1.0
  - New Features
    * Added Transfer prefix in transactional APIs.
- 2.0.0
  - Initial version.

EDMA

The current eDMA driver version is 2.4.4.

- 2.4.4
  - Bug Fixes
    - Fixed comments by replacing STCD with TCD
    - Fixed the TCD overwrite issue when submit transfer request in the callback if there is a active TCD in hardware.
    - Fixed violations of MISRA C-2012 rule 10.8,5.6.
- 2.4.3
  - Improvements
    - Added FSL_FEATURE_MEMORY_HAS_ADDRESS_OFFSET to convert the address between system mapped address and dma quick access address.
  - Bug Fixes
    - Fixed the wrong tcd done count calculated in first TCD interrupt for the non scatter gather case.
- 2.4.2
  - Bug Fixes
    - Fixed the wrong tcd done count calculated in first TCD interrupt by correct the initial value of the header.
    - Fixed violations of MISRA C-2012 rule 10.3, 10.4.
- 2.4.1
  - Bug Fixes
    - Added clear CITER and BITER registers in EDMA_AbortTransfer to make sure the TCD registers in a correct state for next calling of EDMA_SubmitTransfer.
    - Removed the clear DONE status for ESG not enabled case to aovid DONE bit cleared unexpectedly.
- 2.4.0
  - Improvements
    - Added api EDMA_EnableContinuousChannelLinkMode to support continuous link mode.
    - Added apis EDMA_SetMajorOffsetConfig/EDMA_TcdSetMajorOffsetConfig to support major loop address offset feature.
    - Added api EDMA_EnableChannelMinorLoopMapping for minor loop offset feature.
    - Removed the reduntant IRQ Handler in edma driver.
- 2.3.2
  - Improvements
    - Fixed HIS ccm issue in function EDMA_PrepareTransferConfig.
    - Fixed violations of MISRA C-2012 rule 11.6, 10.7, 10.3, 18.1.
  - Bug Fixes
    - Added ACTIVE & BITER & CITER bitfields to determine the channel status to fixed the issue of the transfer request cannot submit by function EDMA_SubmitTransfer when channel is idle.
- 2.3.1
  - Improvements

    * Added source/destination address alignment check.
    * Added driver IRQ handler support for multi DMA instance in one SOC.
- 2.3.0
  - Improvements
    * Added new api EDMA_PrepareTransferConfig to allow different configurations of width and offset.
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.4, 10.1.
    * Fixed the Coverity issue regarding out-of-bounds write.
- 2.2.0
  - Improvements
    * Added peripheral-to-peripheral support in EDMA driver.
- 2.1.9
  - Bug Fixes
    * Fixed MISRA issue: Rule 10.7 and 10.8 in function EDMA_DisableChannelInterrupts and EDMA_SubmitTransfer.
    * Fixed MISRA issue: Rule 10.7 in function EDMA_EnableAsyncRequest.
- 2.1.8
  - Bug Fixes
    * Fixed incorrect channel preemption base address used in EDMA_SetChannelPreemption-Config API which causes incorrect configuration of the channel preemption register.
- 2.1.7
  - Bug Fixes
    * Fixed incorrect transfer size setting.
      · Added 8 bytes transfer configuration and feature for RT series;
      · Added feature to support 16 bytes transfer for Kinetis.
    * Fixed the issue that EDMA_HandleIRQ would go to incorrect branch when TCD was not used and callback function not registered.
- 2.1.6
  - Bug Fixes
    * Fixed KW3X MISRA Issue.
      · Rule 14.4, 10.8, 10.4, 10.7, 10.1, 10.3, 13.5, and 13.2.
  - Improvements
    * Cleared the IRQ handler unavailable for specific platform with macro FSL_FEATURE_-EDMA_MODULE_CHANNEL_IRQ_ENTRY_SHARED_OFFSET.
- 2.1.5
  - Improvements
    * Improved EDMA IRQ handler to support half interrupt feature.
- 2.1.4
  - Bug Fixes
    * Cleared enabled request, status during EDMA_Init for the case that EDMA is halted before reinitialization.
- 2.1.3
  - Bug Fixes
    * Added clear DONE bit in IRQ handler to avoid overwrite TCD issue.

    * Optimized above solution for the case that transfer request occurs in callback.
- 2.1.2
  - Improvements
    * Added interface to get next TCD address.
    * Added interface to get the unused TCD number.
- 2.1.1
  - Improvements
    * Added documentation for eDMA data flow when scatter/gather is implemented for the EDMA_HandleIRQ API.
    * Updated and corrected some related comments in the EDMA_HandleIRQ API and edma_handle_t struct.
- 2.1.0
  - Improvements
    * Changed the EDMA_GetRemainingBytes API into EDMA_GetRemainingMajorLoopCount due to eDMA IP limitation (see API comments/note for further details).
- 2.0.5
  - Improvements
    * Added pubweak DriverIRQHandler for K32H844P (16 channels shared).
- 2.0.4
  - Improvements
    * Added support for SoCs with multiple eDMA instances.
    * Added pubweak DriverIRQHandler for KL28T DMA1 and MCIMX7U5_M4.
- 2.0.3
  - Bug Fixes
    * Fixed the incorrect pubweak IRQHandler name issue, which caused re-definition build errors when client set his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler.
- 2.0.2
  - Bug Fixes
    * Fixed incorrect minorLoopBytes type definition in _edma_transfer_config struct, and defined minorLoopBytes as uint32_t instead of uint16_t.
- 2.0.1
  - Bug Fixes
    * Fixed the eDMA callback issue (which did not check valid status) in EDMA_HandleIRQ API.
- 2.0.0
  - Initial version.

EWM

The current EWM driver version is 2.0.3.

- 2.0.3
  - Bug Fixes

    - Fixed violation of MISRA C-2012 rules: 10.1, 10.3.
- 2.0.2
  - Bug Fixes
    - Fixed violation of MISRA C-2012 rules: 10.3, 10.4.
- 2.0.1
  - Bug Fixes
    - Fixed the hard fault in EWM_Deinit.
- 2.0.0
  - Initial version.

FLASH

Current FLASH driver version is 3.1.2

- 3.1.2
  - Bug Fixes — Remove redundant comments.
- 3.1.1
  - Bug Fixes — MISRA C-2012 issue fixed: rule 10.3
- 3.1.0
  - New Feature
    - Support erase flash asynchronously.
- 3.0.2
  - Bug Fixes — MISRA C-2012 issue fixed: rule 8.4, 17.7, 10.4, 16.1, 21.15, 11.3, 10.7 — building warning -Wnull-dereference on arm compiler v6
- 3.0.1
  - New Features
    - Added support FlexNVM alias for (kw37/38/39).
- 3.0.0
  - Improvements
    - Reorganized FTFx flash driver source file.
    - Extracted flash cache driver from FTFx driver.
    - Extracted flexnvm flash driver from FTFx driver.
- 2.3.1
  - Bug Fixes
    - Unified Flash IFR design from K3.
    - New encoding rule for K3 flash size.
- 2.3.0
  - New Features
    - Added support for device with LP flash (K3S/G).
    - Added flash prefetch speculation APIs.
  - Improvements
    - Refined flash_cache_clear function.
    - Reorganized the member of flash_config_t struct.
- 2.2.0

  - New Features
    - Supported FTFL device in FLASH_Swap API.
    - Supported various pflash start addresses.
    - Added support for KV58 in cache clear function.
    - Added support for device with secondary flash (KW40).
  - Bug Fixes
    - Compiled execute-in-ram functions as PIC binary code for driver use.
    - Added missed flexram properties.
    - Fixed unaligned variable issue for execute-in-ram function code array.
- 2.1.0
  - Improvements
    - Updated coding style to align with KSDK 2.0.
    - Different-alignment-size support for pflash and flexnvm.
    - Improved the implementation of execute-in-ram functions.
- 2.0.0
  - Initial version

FTM

The current FTM driver version is 2.5.0.

- 2.5.0
  - Improvements
    - Added FTM_CalculateCounterClkDiv to help calculates the counter clock prescaler.
    - Modify FTM_UpdatePwmDutycycle API to make it return pwm duty cycles status.
  - Bug Fixes
    - Fixed TPM_SetupPwm can't configure 100% center align combined PWM issues.
- 2.4.1
  - Bug Fixes
    - Added function macro to determine if FTM instance has only basic features, to prevent access to protected register bits.
- 2.4.0
  - Improvements
    - Added CNTIN register initialization in FTM_SetTimerPeriod API.
    - Added a new API to read the captured value of a FTM channel configured in capture mode:
      - FTM_GetInputCaptureValue()
- 2.3.0
  - Improvements
    - Added support of EdgeAligned/CenterAligned/Asymmetrical combine PWM mode in F-TM_SetupPWM() and FTM_SetupPwmMode() APIs.
    - Remove kFTM_ComplementaryPwm from support PWM mode, and add new parameter "enableComplementary" in structure ftm_chnl_pwm_signal_param_t.
    - Rename FTM_SetupFault() API to FTM_SetupFaultInput() to avoid ambiguity.

- 2.2.3
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 14.4 and 17.7.
- 2.2.2
  - Bug Fixes
    - Fixed the issue that when FTM instance has only TPM features cannot be initialized by FTM_Init() function. By added function macro to assert FTM is TPM only instance.
- 2.2.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.1, 10.3, 10.4, 10.6, 10.7 and 11.9.
- 2.2.0
  - Bug Fixes
    - Fixed the issue of comparison between signed and unsigned integer expressions.
  - Improvements
    - Added support of complementary mode in FTM_SetupPWM() and FTM_SetupPwm-Mode() APIs.
    - Added new parameter "enableDeadtime" in structure ftm_chnl_pwm_signal_param_t.
- 2.1.1
  - Bug Fixes
    - Fixed COVERITY integer handing issue where the right operand of a left bit shift statement should not be a negative value. This appears in FTM_SetReloadPoints().
- 2.1.0
  - Improvements
    - Added a new API FTM_SetupPwmMode() to allow the user to set the channel match value in units of timer ticks. New configure structure called ftm_chnl_pwm_config_-param_t was added to configure the channel's PWM parameters. This API is similar with FTM_SetupPwm() API, but the new API will not set the timer period(MOD value), it will be useful for users to set the PWM parameters without changing the timer period.
  - Bug Fixes
    - Added feature macro to enable/disable the external trigger source configuration.
- 2.0.4
  - Improvements
    - Added a new API to enable DMA transfer:
      - FTM_EnableDmaTransfer()
- 2.0.3
  - Bug Fixes
    - Updated the FTM driver to enable fault input after configuring polarity.
- 2.0.2
  - Improvements
    - Added support to Quad Decoder feature with new APIs:
      - FTM_GetQuadDecoderFlags()
      - FTM_SetQuadDecoderModuloValue()
      - FTM_GetQuadDecoderCounterValue()
      - FTM_ClearQuadDecoderCounterValue()
- 2.0.1

  - Bug Fixes
    - Updated the FTM driver to fix write to ELSA and ELSB bits.
    - FTM combine mode: set the COMBINE bit before writing to CnV register.
- 2.0.0
  - Initial version.

GPIO

The current driver version is 2.6.0.

- 2.6.0
  - New Features
    - Added API to get GPIO version information.
    - Added API to control a pin for general purpose input.
    - Added some APIs to control pin in secure and previliege status.
- 2.5.3
  - Bug Fixes
    - Correct the feature macro typo: FSL_FEATURE_GPIO_HAS_NO_INDEP_OUTPUT_-CONTORL.
- 2.5.2
  - Improvements
    - Improved GPIO_PortSet/GPIO_PortClear/GPIO_PortToggle functions to support devices without Set/Clear/Toggle registers.
- 2.5.1
  - Bug Fixes
    - Fixed wrong macro definition.
    - Fixed MISRA C-2012 rule issues in the FGPIO_CheckAttributeBytes() function.
    - Defined the new macro to separate the scene when the width of registers is different.
    - Removed some redundant macros.
  - New Features
    - Added some APIs to get/clear the interrupt status flag when the port doesn't control pins' interrupt.
- 2.4.1
  - Improvements
    - Improved GPIO_CheckAttributeBytes() function to support 8 bits width GACR register.
- 2.4.0
  - Improvements
    - API interface added:
      - New APIs were added to configure the GPIO interrupt clear settings.
- 2.3.2
  - Bug Fixes
    - Fixed the issue for MISRA-2012 check.
      - Fixed rule 3.1, 10.1, 8.6, 10.6, and 10.3.
- 2.3.1

- Improvements
  - Removed deprecated APIs.
- 2.3.0
  - New Features
    - Updated the driver code to adapt the case of interrupt configurations in GPIO module. New APIs were added to configure the GPIO interrupt settings if the module has this feature on it.
- 2.2.1
  - Improvements
    - API interface changes:
      - Refined naming of APIs while keeping all original APIs by marking them as deprecated. The original APIs will be removed in next release. The main change is updating APIs with prefix of _PinXXX() and _PortXXX.
- 2.1.1
  - Improvements
    - API interface changes:
      - Added an API for the check attribute bytes.
- 2.1.0
  - Improvements
    - API interface changes:
      - Added "pins" or "pin" to some APIs' names.
      - Renamed "_PinConfigure" to "GPIO_PinInit".

I2C

The current I2C driver version is 2.0.9.

- 2.0.9
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 10.1, 10.4, 13.5, 20.8.
- 2.0.8
  - Bug Fixes
    - Fixed the bug that DFEN bit of I2C Status register 2 could not be set in I2C_MasterInit.
    - MISRA C-2012 issue fixed: rule 14.2, 15.7, and 16.4.
    - Eliminated IAR Pa082 warnings from I2C_MasterTransferDMA and I2C_MasterTransferCallbackDMA by assigning volatile variables to local variables and using local variables instead.
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 11.9, 14.4, 15.7, 17.7.
  - Improvements
    - Improved timeout mechanism when waiting certain state in transfer API.
    - Updated the I2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
    - Moved the master manually acknowledge byte operation into static function I2C_Master-

    AckByte.
    * Fixed control/status clean flow issue inside I2C_MasterReadBlocking to avoid potential issue that pending status is cleaned before it's proceeded.
- 2.0.7
  - Bug Fixes
    * Fixed the issue for MISRA-2012 check.
      · Fixed rule 11.9 ,15.7 ,14.4 ,10.4 ,10.8 ,10.3, 10.1, 10.6, 13.5, 11.3, 13.2, 17.7, 5.7, 8.3, 8.5, 11.1, 16.1.
    * Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
    * Fixed variable redefine issue by moving i2cBases from fsl_i2c.h to fsl_i2c.c.
  - Improvements
    * Added I2C_MASTER_FACK_CONTROL macro to enable FACK control for master transfer receive flow with IP supporting double buffer, then master could hold the SCL by manually setting TX AK/NAK during data transfer.
- 2.0.6
  - Bug Fixes
    * Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) did not support the situation of master transfer with subaddress and transfer data size being zero, which means no data followed by the subaddress.
- 2.0.5
  - Improvements
    * Added I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.4
  - Bug Fixes
    * Added a proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Support write only or write+read with no start flag; does not support read only with no start flag.
- 2.0.3
  - Bug Fixes
    * Removed enableHighDrive member in the master/slave configuration structure because the operation to HDRS bit is useless, the user need to use DSE bit in port register to configure the high drive capability.
    * Added register reset operation in I2C_MasterInit and I2C_SlaveInit APIs. Fixed issue where I2C could not switch between master and slave mode.
    * Improved slave IRQ handler to handle the corner case that stop flag and address match flag come synchronously.
- 2.0.2
  - Bug Fixes
    * Fixed issue in master receive and slave transmit mode with no stop flag. The master could not succeed to start next transfer because the master could not send out re-start signal.
    * Fixed the out-of-order issue of data transfer due to memory barrier.
    * Added hold time configuration for slave. By leaving the SCL divider and MULT reset values when configured to slave mode, the setup and hold time of the slave is then reduced outside of spec for lower baudrates. This can cause intermittent arbitration loss on the

master side.
  - New Features
    - Added address nak event for master.
    - Added general call event for slave.
- 2.0.1
  - New Features
    - Added double buffer enable configuration for SoCs which have the DFEN bit in S2 register.
    - Added flexible transmit/receive buffer size support in I2C_SlaveHandleIRQ.
    - Added start flag clear, address match, and release bus operation in I2C_SlaveWrite/ReadBlocking API.
  - Bug Fixes
    - Changed the kI2C_SlaveRepeatedStartEvent to kI2C_SlaveStartEvent.
- 2.0.0
  - Initial version.

LLWU

The current LLWU driver version is 2.0.5.

- 2.0.5
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3.
    - Fixed the issue that function LLWU_SetExternalWakeupPinMode() does not work on 32-bit width platforms.
- 2.0.4
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 10.6, 10.7, 11.3.
    - Fixed issue that LLWU_ClearExternalWakeupPinFlag may clear other filter flags by mistake on platforms with 32-bit LLWU registers.
- 2.0.3
  - Bug Fixes
    - Fixed MISRA-2012 rules.
      - Rule 16.4.
- 2.0.2
  - Improvements
    - Corrected driver function LLWU_SetResetPinMode parameter name.
  - Bug Fixes
    - Fixed MISRA-2012 rules.
      - Rule 14.4, 10.8, 10.4, 10.3.
- 2.0.1
  - Other Changes
    - Updates for KL8x.
- 2.0.0

- Initial version.

LPTMR

The current LPTMR driver version is 2.1.1.

- 2.1.1
  - Improvements
    - Updated the characters from "PTMR" to "LPTMR" in "FSL_FEATURE_PTMR_HAS_-NO_PRESCALER_CLOCK_SOURCE_1_SUPPORT" feature definition.
- 2.1.0
  - Improvements
    - Implement for some special devices' not supporting for all clock sources.
  - Bug Fixes
    - Fixed issue when accessing CMR register.
- 2.0.2
  - Bug Fixes
    - Fixed MISRA-2012 issues.
      - Rule 10.1.
- 2.0.1
  - Improvements
    - Updated the LPTMR driver to support 32-bit CNR and CMR registers in some devices.
- 2.0.0
  - Initial version.

LPUART

The current LPUART driver version is 2.5.3.

- 2.5.3
  - Bug Fixes
    - Fixed comments by replacing unused status flags kLPUART_NoiseErrorInRxDataReg-Flag and kLPUART_ParityErrorInRxDataRegFlag with kLPUART_NoiseErrorFlag and kLPUART_ParityErrorFlag.
- 2.5.2
  - Bug Fixes
    - Fixed bug that when setting watermark for TX or RX FIFO, the value may exceed the maximum limit.
  - Improvements
    - Added check in LPUART_TransferDMAHandleIRQ and LPUART_TransferEdma-HandleIRQ to ensure if user enables any interrupts other than transfer complete interrupt, the dma transfer is not terminated by mistake.
- 2.5.1
  - Improvements

    * Use separate data for TX and RX in lpuart_transfer_t.
  - Bug Fixes
    * Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling LPUART_TransferReceiveNonBlocking, the received data count returned by LPUART_TransferGetReceiveCount is wrong.
- 2.5.0
  - Bug Fixes
    * Added missing interrupt enable masks kLPUART_Match1InterruptEnable and kLPUART_Match2InterruptEnable.
    * Fixed bug in LPUART_EnableInterrupts, LPUART_DisableInterrupts and LPUART_GetEnabledInterrupts that the BAUD[LBKDIE] bit field should be soc specific.
    * Fixed bug in LPUART_TransferHandleIRQ that idle line interrupt should be disabled when rx data size is zero.
    * Deleted unused status flags kLPUART_NoiseErrorInRxDataRegFlag and kLPUART_ParityErrorInRxDataRegFlag, since firstly their function are the same as kLPUART_NoiseErrorFlag and kLPUART_ParityErrorFlag, secondly to obtain them one data word must be read out thus interfering with the receiving process.
    * Fixed bug in LPUART_GetStatusFlags that the STAT[LBKDIF], STAT[MA1F] and STAT[MA2F] should be soc specific.
    * Fixed bug in LPUART_ClearStatusFlags that tx/rx FIFO is reset by mistake when clearing flags.
    * Fixed bug in LPUART_TransferHandleIRQ that while clearing idle line flag the other bits should be masked in case other status bits be cleared by accident.
    * Fixed bug of race condition during LPUART transfer using transactional APIs, by disabling and re-enabling the global interrupt before and after critical operations on interrupt enable register.
    * Fixed DMA/eDMA transfer blocking issue by enabling tx idle interrupt after DMA/eDMA transmission finishes.
  - New Features
    * Added APIs LPUART_GetRxFifoCount/LPUART_GetTxFifoCount to get rx/tx FIFO data count.
    * Added APIs LPUART_SetRxFifoWatermark/LPUART_SetTxFifoWatermark to set rx/tx FIFO water mark.
- 2.4.1
  - Bug Fixes
    * Fixed MISRA advisory 17.7 issues.
- 2.4.0
  - New Features
    * Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.1
  - Bug Fixes
    * Fixed MISRA advisory 15.5 issues.
- 2.3.0
  - Improvements
    * Modified LPUART_TransferHandleIRQ so that txState will be set to idle only when all

data has been sent out to bus.
    - Modified LPUART_TransferGetSendCount so that this API returns the real byte count that LPUART has sent out rather than the software buffer status.
    - Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.8
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule-10.3, rule-14.4, rule-15.5.
    - Eliminated Pa082 warnings by assigning volatile variables to local variables and using local variables instead.
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 10.8, 14.4, 11.6, 17.7.
  - Improvements
    - Added check for kLPUART_TransmissionCompleteFlag in LPUART_WriteBlocking, LPUART_TransferHandleIRQ, LPUART_TransferSendDMACallback and LPUART_-SendEDMACallback to ensure all the data would be sent out to bus.
    - Rounded up the calculated sbr value in LPUART_SetBaudRate and LPUART_Init to achieve more acurate baudrate setting. Changed osr from uint32_t to uint8_t since osr's bigest value is 31.
    - Modified LPUART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
- 2.2.7
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule-12.1, rule-17.7, rule-14.4, rule-13.3, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3.
- 2.2.6
  - Bug Fixes
    - Fixed the issue of register's being in repeated reading status while dealing with the IRQ routine.
- 2.2.5
  - Bug Fixes
    - Do not set or clear the TIE/RIE bits when using LPUART_EnableTxDMA and LPUAR-T_EnableRxDMA.
- 2.2.4
  - Improvements
    - Added hardware flow control function support.
    - Added idle-line-detecting feature in LPUART_TransferNonBlocking function. If an idle line is detected, a callback is triggered with status kStatus_LPUART_IdleLineDetected returned. This feature may be useful when the received Bytes is less than the expected received data size. Before triggering the callback, data in the FIFO (if has FIFO) is read out, and no interrupt will be disabled, except for that the receive data size reaches 0.
    - Enabled the RX FIFO watermark function. With the idle-line-detecting feature enabled, users can set the watermark value to whatever you want (should be less than the RX FIFO size). Data is received and a callback will be triggered when data receive ends.

- 2.2.3
  - Improvements
    - Changed parameter type in LPUART_RTOS_Init struct from rtos_lpuart_config to lpuart-_rtos_config_t.
  - Bug Fixes
    - Disabled LPUART receive interrupt instead of all NVICs when reading data from ring buffer. Otherwise when the ring buffer is used, receive nonblocking method will disable all NVICs to protect the ring buffer. This may has a negative effect on other IPs that are using the interrupt.
- 2.2.2
  - Improvements
    - Added software reset feature support.
    - Added software reset API in LPUART_Init.
- 2.2.1
  - Improvements
    - Added separate RX/TX IRQ number support.
- 2.2.0
  - Improvements
    - Added support of 7 data bits and MSB.
- 2.1.1
  - Improvements
    - Removed unnecessary check of event flags and assert in LPUART_RTOS_Receive.
    - Added code to always wait for RX event flag in LPUART_RTOS_Receive.
- 2.1.0
  - Improvements
    - Update transactional APIs.

LPUART_EDMA

The current LPUART_EDMA driver version is 2.4.0.

- 2.4.0
  - Refer LPUART driver change log 2.1.0 to 2.4.0

LPUART_FREERTOS

The current LPUART_FREERTOS driver version is 2.4.0.

- 2.4.0
  - Refer LPUART driver change log 2.1.0 to 2.4.0

PDB

The current PDB driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.1 and 10.4.
- 2.0.3
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 17.7.
- 2.0.2
  - Improvement:
    - Used macros in feature file instead of that in IO map.
- 2.0.1
  - Changed PDB register base array to const.
- 2.0.0
  - Initial version.

PIT

The current PIT driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed PIT_SetTimerPeriod implementation, the load value trigger should be PIT clock cycles minus 1.
- 2.0.3
  - Bug Fixes
    - Clear all status bits for all channels to make sure the status of all TCTRL registers is clean.
- 2.0.2
  - Bug Fixes
    - Fixed MISRA-2012 issues.
      - Rule 10.1.
- 2.0.1
  - Bug Fixes
    - Cleared timer enable bit for all channels in function PIT_Init() to make sure all channels stay in disable status before setting other configurations.
    - Fixed MISRA-2012 rules.
      - Rule 14.4, rule 10.4.
- 2.0.0
  - Initial version.

PMC

The current PMC driver version is 2.0.3.

- 2.0.3
  - Bug Fixes
    - Fixed the violation of MISRA C-2012 rule 11.3.
- 2.0.2
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules:
      - Rule 10.3.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA issues.
      - Rule 10.8, Rule 10.3.
- 2.0.0
  - Initial version.

PORT

The current PORT driver version is 2.3.0.

- 2.3.0
  - New Features
    - Added new APIs for Electrical Fast Transient(EFT) detect.
    - Added new API to configure port voltage range.
- 2.2.0
  - New Features
    - Added new api PORT_EnablePinDoubleDriveStrength.
- 2.1.1
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules: 10.1, 10.411.311.8, 14.4.
- 2.1.0
  - New Features
    - Updated the driver code to adapt the case of the interrupt configurations in GPIO module. Will move the pin configuration APIs to GPIO module.
- 2.0.2
  - Other Changes
    - Added feature guard macros in the driver.
- 2.0.1
  - Other Changes
    - Added "const" in function parameter.
    - Updated some enumeration variables’ names.

RCM

The current RCM driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed violation of MISRA C-2012 rule 10.3
- 2.0.3
  - Bug Fixes
    - Fixed violation of MISRA C-2012 rules.
- 2.0.2
  - Bug Fixes
    - Fixed MISRA issue.
      - Rule 10.8, rule 10.1, rule 13.2, rule 3.1.
- 2.0.1
  - Bug Fixes
    - Fixed kRCM_SourceSw bit shift issue.
- 2.0.0
  - Initial version.

SIM

The current SIM driver version is 2.1.3.

- 2.1.3
  - Improvements
    - Updated function SIM_GetUniqueId to support different register names.
- 2.1.2
  - Bug Fixes
    - Fixed SIM_GetUniqueId bug that could not get UIDH.
- 2.1.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.4
- 2.1.0
  - Improvements
    - Added new APIs: SIM_GetRfAddr() and SIM_EnableSystickClock().
- 2.0.0
  - Initial version.

SMC

The current SMC driver version is 2.0.7.

- 2.0.7
  - Bug Fixes

    - Fixed MISRA-2012 issue 10.3.
- 2.0.6
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule 10.3, rule 11.3.
- 2.0.5
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule 15.7, rule 14.4, rule 10.3, rule 10.1, rule 10.4.
- 2.0.4
  - Bug Fixes
    - When entering stop modes, used RAM function for the flash synchronization issue. Application should make sure that, the RW data of fsl_smc.c is located in memory region which is not powered off in stop modes.
- 2.0.3
  - Improvements
    - Added APIs SMC_PreEnterStopModes, SMC_PreEnterWaitModes, SMC_PostExitWait-Modes, and SMC_PostExitStopModes.
- 2.0.2
  - Bug Fixes
    - Added DSB before WFI while ISB after WFI.
  - Other Changes
    - Updated SMC_SetPowerModeVlpw implementation.
- 2.0.1
  - Other Changes
    - Updated for KL8x.
- 2.0.0
  - Initial version.

UART

The current UART driver version is 2.5.1.

- 2.5.1
  - Improvements
    - Use separate data for TX and RX in uart_transfer_t.
  - Bug Fixes
    - Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling UART_TransferReceiveNonBlocking, the received data count returned by UAR-T_TransferGetReceiveCount is wrong.
- 2.5.0
  - New Features
    - Added APIs UART_GetRxFifoCount/UART_GetTxFifoCount to get rx/tx FIFO data count.

    - Added APIs UART_SetRxFifoWatermark/UART_SetTxFifoWatermark to set rx/tx FIFO water mark.
  - Bug Fixes
    - Fixed bug of race condition during UART transfer using transactional APIs, by disabling and re-enabling the global interrupt before and after critical operations on interrupt enable registers.
    - Fixed DMA/eDMA transfer blocking issue by enabling tx idle interrupt after DMA/eDM-A transmission finishes.
- 2.4.0
  - New Features
    - Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.0
  - Bug Fixes
    - Fixed the bug that, when framing/parity/noise/overflow flag or idle line detect flag is set, receive FIFO should be flushed to avoid FIFO pointer being in unknown state, since FIFO has no valid data.
  - Improvements
    - Modified UART_TransferHandleIRQ so that txState will be set to idle only when all data has been sent out to bus.
    - Modified UART_TransferGetSendCount so that this API returns the real byte count that UART has sent out rather than the software buffer status.
    - Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.0
  - New Features
    - Added UART hardware FIFO enable/disable API.
  - Improvements
    - Added check for kUART_TransmissionCompleteFlag in UART_TransferHandleIRQ, U-ART_SendEDMACallback and UART_TransferSendDMACallback to ensure all the data would be sent out to bus.
  - Bug Fixes
    - Eliminated IAR Pa082 warnings from UART_TransferGetRxRingBufferLength, UART-_GetEnabledInterrupts, UART_GetStatusFlags and UART_TransferHandleIRQ.
    - Added code in UART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 14.4, 11.6, 17.7.
- 2.1.6
  - Bug Fixes
    - Fixed the issue of register's being in repeatedly reading status while performing the IRQ routine.
- 2.1.5
  - Improvements
    - Added hardware flow control function support.
    - Added idle-line-detecting feature in UART_TransferNonBlocking function. If an idle line is detected, a callback will be triggered with status kStatus_UART_IdleLineDetected

returned. This feature may be useful when the number of received bytes is less than the expected receive data size. Before triggering the callback, data in the FIFO is read out (if it has FIFO), and no interrupt will be disabled except for the case that the receive data size reaches 0.
    * Enabled the RX FIFO watermark function. With the idle-line-detecting feature enabled, you can set the watermark value to whatever you want (should not be bigger than the RX FIFO size). Data is then received and a callback will be triggered when data receive ends.
- 2.1.4
  - Improvements
    * Changed parameter type in UART_RTOS_Init() struct rtos_uart_config –> uart_rtos_-config_t.
  - Bug Fixes
    * Disabled UART receive interrupt instead of global interrupt when reading data from ring buffer. With ring buffer used, receive nonblocking will disable global interrupt to protect the ring buffer. This has a negative effect on other IPs using interrupt.
- 2.1.3
  - New Features
    * Added RX framing error and parity error status check when using interrupt transfer.
- 2.1.2
  - Bug Fixes
    * Fixed baud rate fine adjust bug to make the computed baud rate more accurate.
- 2.1.1
  - Bug Fixes
    * Removed needless check of event flags and assert in UART_RTOS_Receive.
    * Always waited for RX event flag in UART_RTOS_Receive.
- 2.1.0
  - Improvements
    * Added transactional API.
- 2.0.0
  - Initial version.

UART_EDMA

The current UART_EDMA driver version is 2.5.2.

- 2.5.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules.
- 2.5.1
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules.
- 2.5.0
  - Refer UART driver change log 2.1.0 to 2.5.0

UART_FREERTOS

The current UART_FREERTOS driver version is 2.5.0.

- 2.5.0
  - Refer UART driver change log 2.1.0 to 2.5.0

VREF

The current VREF driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    - Fixed the violation of MISRA-2012 rule 10.3.
    - Fixed MISRA C-2012 rule 10.3, rule 10.4 violation.
- 2.1.1
  - Bug Fixes
    - MISRA-2012 issue fixed.
      - Fixed rules containing: rule-10.4, rule-10.3, rule-10.1.
- 2.1.0
  - Improvements
    - Added new functions to support L5K board: added VREF_SetTrim2V1Val() and VREF-_GetTrim2V1Val() functions to supply 2V1 output mode.
- 2.0.0
  - Initial version.

WDOG

The current WDOG driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3, 10.4, 10.6, 10.7 11.9 and 17.7.
- 2.0.0
  - Initial version.

2 Middleware Change Log

FatFs for MCUXpresso SDK

Current version is FatFs R0.14b_rev0.

- R0.14b_rev0
  - Upgraded to version 0.14b
- R0.14a_rev0
  - Upgraded to version 0.14a
  - Applied patch ff14a_p1.diff and ff14a_p2.diff
- R0.14_rev0
  - Upgraded to version 0.14
  - Applied patch ff14_p1.diff and ff14_p2.diff
- R0.13c_rev0
  - Upgraded to version 0.13c
  - Applied patches ff_13c_p1.diff,ff_13c_p2.diff, ff_13c_p3.diff and ff_13c_p4.diff.
- R0.13b_rev0
  - Upgraded to version 0.13b
- R0.13a_rev0
  - Upgraded to version 0.13a. Added patch ff_13a_p1.diff.
- R0.12c_rev1
  - Add NAND disk support.
- R0.12c_rev0
  - Upgraded to version 0.12c and applied patches ff_12c_p1.diff and ff_12c_p2.diff.
- R0.12b_rev0
  - Upgraded to version 0.12b.
- R0.11a
  - Added glue functions for low-level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c.
  - Added RTOS wrappers to make FatFs thread safe. Modified syscall.c.
  - Renamed ffconf.h to ffconf_template.h. Each application should contain its own ffconf.h.
  - Included ffconf.h into diskio.c to enable the selection of physical disk from ffconf.h by macro definition.
  - Conditional compilation of physical disk interfaces in diskio.c.

FreeMASTER Communication Driver

Current version is 3.0.4. Visit `https://www.nxp.com/freemaster` for more information. Reach out for a support at `https://community.nxp.com/community/freemaster`.

- 3.0.0
  - Initial version of FreeMASTER driver reworked from a standalone package to MCUXpresso SDK middleware.
  - This driver version supports new version V4 of FreeMASTER serial communication protocol.

  - Supports UART, LPUART, USART, MINIUSART, FlexCAN, USB-CDC and JTAG/BDM communication.
  - Initial version was tested with the following boards: evkmimxrt1060, frdmk64f, frdmke15z, frdmkl28z, lpcxpresso54628 lpcxpresso55s69, lpcxpresso845max and twrk64f120m.
  - Use with FreeMASTER PC Host tool version 2.5 or later.
- 3.0.1
  - FreeMASTER driver extended to support wide range of Kinetis, LPC and i.MX-RT platforms.
  - Low-level communication drivers also available for few non-SDK NXP platforms like S12Z, S32x and more.
  - Use with FreeMASTER PC Host tool version 3.0 or later.
- 3.0.2
  - FreeMASTER driver support of DSC56F800EX and S12 platforms extended.
  - Removed dependency on C99 compiler features.
  - Use with FreeMASTER PC Host tool version 3.0.2 or later.
- 3.0.3
  - General update for SDK 2.9.0
  - fmstr_any demo added to selected platforms - use with MCUXpresso SDK and FreeMASTER peripheral configuration tool.
  - New example.pmp project file embedded into application flash storage.
  - USB-CDC implementation fixed, new JTAG EOnCE communication interface added to DSC 56F800E family.
  - Use with FreeMASTER PC Host tool version 3.0.3 or later. Version 3.1.x is recommended.
- 3.0.4
  - Fixed component dependency logic of FreeMASTER driver.
  - Use with FreeMASTER PC Host tool version 3.1.x
- 3.0.5
  - General update for SDK 2.11 and 2.12
  - New TCP and UDP support with lwIP stack
  - New communication over Segger RTT interface
  - Add fmstr_net and fmstr_wifi examples for selected i.MX-RT platforms
  - Add fmstr_rtt example for selected platforms
  - Fixed negative recorder threshold trigger processing

MOTOR_CONTROL for KSDK

Current version is 1.1.0

- 1.1.0
  - Initial version.

RTCESL for KSDK

Current version is 4.3

- 4.3
  - Initial version.

SAFETY_IEC60730B for KSDK

Current version is 1.1.0

- 1.1.0
  - Initial version.

SD(SPI) Card driver for MCUXpresso SDK

The current driver version is 2.2.1.

- 2.2.1
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.2.0
  - Improvements
    * Added init/deinit/csActivePolarity function pointer in sdspi_host_t.
    * Added retry function in SDSPI_SetBlockSize to fix the issue that set block size command failed found on low capability card.
    * Added internalBuffer in sdspi_card_t instead if use rawcid/rawcsd/rawscr to reduce the memoery cost of card descriptor.
- 2.1.4
  - Bug Fixes
    * Fixed MDK 66-D warning.
- 2.1.3
  - Improvements
    * Improved sdspi code size and performance.
- 2.0.0
  - Initial version.

3 Component Change Log

SERIAL_MANAGER

The current Serial_Manager component version is 1.0.2.

- 1.0.2
  - Add SerialManager_WriteTimeDelay()/SerialManager_ReadTimeDelay() for serial manager's read/write non-blocking mode.
- 1.0.1
  - Add prefixing fsl_component_xxx/fsl_adapter_xxx.
- 1.0.0
  - Initial version