MCUXpresso SDK Release Notes Supporting evkmimxrt1010

Change Logs

NXP Semiconductors

Contents

Driver Change Log

Title

Page No.

Middleware Change Log

Title

Page No.

Component Change Log

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.5.3.

- 2.5.3
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rule 10.7.
- 2.5.2
  - Bug Fixes
    - Fixed issues in CLOCK_GetSysPfdFreq() and CLOCK_GetUsb1PfdFreq() which produce incorrect result.
- 2.5.1
  - Improvements
    - Added enumeration clock_div_value_t.
- 2.5.0
  - New Features
    - Added CLOCK_IsUsb1PfdEnabled and CLOCK_IsSysPfdEnabled to get the clock source status.
  - Bug Fixes
    - Fixed the wrong mxu enumerator used in "CLOCK_ROOT_MUX_TUPLE" macro.
    - Updated CLOCK_SetClockOutput1() function, adjusted the sequence of register configuration.
- 2.4.1
  - Improvements
    - Placed function internal constants into initialized data segment.
- 2.4.0
  - New Features
    - Added clock output related APIs and data structures.
    - Added one function CLOCK_GetClockRootFreq to get the frequency of each clock root.
- 2.3.1
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rule10.1, rule 18.4, rule 14.4 and so on.
- 2.3.0
  - New feature:
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.2.0
  - New feature
  - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.1.6
  - Bug Fix:
    - Fix build issue with GCC compiler when include header from C++ file.

- 2.1.5
  - Bug Fix:
    - Add initialization of the fractional mode and spread spectrum mode in CLOCK_InitSysPll().
- 2.1.4
  - Optimization:
    - Add PerClk in clock_name_t and CLOCK_GetFreq.
    - Add APIs to get the frequency of AHB clock and SEMC, IPG clock and PER clock.
- 2.1.3
  - Use double instead of uint64_t to achieve better performance with double precision FPU.
- 2.1.2
  - some minor fixes.
- 2.0.0
  - initial version.

IOMUXC

The current IOMUXC driver version is 2.0.2.

- 2.0.2
  - Doxygen improvement.
- 2.0.1
  - Delete enum value kIOMUXC_GPR_USBExposureMode in the _iomuxc_gpr_mode.
- 2.0.0
  - initial version.

LPI2C_CMSIS

Current LPI2C_CMSIS driver version is 2.5

- 2.5
  - Bug Fixes
    - Fixed MISRA rule 10.3 in LPI2Cx_EdmaResource, use uint16_t force the essential Type.
- 2.4
  - Bug Fixes
    - Fixed rule 10.3 in LPI2C_Master_EdmaInitialize.
- 2.3
  - Imporvement
    - Changed DMA_Type to void for different platform dma.
- 2.2
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 10.3.
- 2.1

  - 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
  - Initial version.

LPSPI_CMSIS

Current LPSPI_CMSIS driver version is 2.9

- 2.9
  - Bug Fixes
    - Fixed rule 10.3 in cmsis_lpspi_edma_resource_t value RTE_SPIx_DMA_TX_PERI_SEL.
- 2.8
  - Bug Fixes
    - Fixed rule 10.3 in LPSPI_EdmaPowerControl.
- 2.7
  - Imporvement
    - Changed DMA_Type to void for different platform dma.
- 2.6
  - Bug Fixes
    - Fixed wrong state busy flag, use the state of Handle instead of RemainingByteCount.
    - Fixed the MISRA-2012 violations.
      - Fixed rule 10.3.
- 2.5
  - Bug Fixes
    - Fixed wrong configuration of setting the bytes to be swapped during transfer when the transfer width is more than 8.
    - Update the edma request source to support more than 0xFF request sources.
- 2.4
  - Bug Fixes
    - Update driver to fix warnings reported by IAR v9.
- 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 incorrect clock polarity assignment in the 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. Please note that this is not supported in slave interrupts, because the pin will stay tristated if tX buffer is NULL.
        - 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.
        - Enabled 3-Wire mode, user can use ARM_SPI_MODE_MASTER_SIMPLEX/ARM_SPI_MODE_SLAVE_SIMPLEX to enable this feature. For ARM_SPI_MODE_MASTER_SIMPLEX mode, the SOUT pin is selected as the input/output pin, and for ARM_SPI_MODE_SLAVE_SIMPLEX, the SIN pin is selected as the input/output pin.
- 2.0
    - Initial version.

LPUART_CMSIS

Current LPUART_CMSIS driver version is 2.6

- 2.6
    - Imporvement
        - Added support of UART6-UART12 in fsl_lpuart_cmsis.c.
- 2.5
    - Imporvement
        - Changed DMA_Type to void for different platform dma.
- 2.4
    - Bug Fixes
        - Fixed the MISRA-2012 violations.
            - Fixed rule 10.3.
- 2.3
    - Other Changes
        - Update the edma request source to support more than 0xFF request sources.
- 2.2
    - Bug Fixes
        - Update driver to fix warnings reported by IAR v9.
- 2.1
    - 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
    - Initial version.

ADC

The current ADC driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed violation of MISRA C-2012 rule 10.4.
- 2.0.3
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules:
      - Rule 10.1 10.4 10.7 17.7.
- 2.0.2
  - Improvements
    - Used conversion control feature macro instead of that in IO map.
- 2.0.1
  - New Features
    - Added a control macro to enable/disable CLOCK code in current driver.
- 2.0.0
  - Initial version.

ADC_ETC

The current ADC_ETC driver version is 2.2.1.

- 2.2.1
  - Improvements
    - Moditied macro "ADC_ETC_DONE2_ERR_IRQ_TRIG0_DONE2_MASK" to "ADC_-ETC_DONE2_3_ERR_IRQ_TRIG0_DONE2_MASK" based on the updates of header file.
- 2.2.0
  - Improvements
    - Defined two macros to support some devices that do not equipped with TSC trigger.
- 2.1.1
  - Bug Fixes
    - Fixed the violation of MISRA-2012 rule.
- 2.1.0
  - New Features
    - Supported independent IRQ enable bit in ADC-ETC chain configuration registers.
    - Supported trigger n DONE3 interrupt operations.
  - Bug Fixes
    - Fixed the violation of MISRA-2012 rules:
      - Rule 10.1 10.3 10.7 15.5 16.1 16.3 16.4 17.7
- 2.0.1
  - New Features
    - Added a control macro to enable/disable the CLOCK code in current driver.

- 2.0.0
  - Initial version.

AIPSTZ

The current AIPSTZ driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3, 10.4, and 14.4.
- 2.0.0
  - Initial version.

AOI

The current AOI driver version is 2.0.2.

- 2.0.2
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.0.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.8, 2.2.
- 2.0.0
  - Initial version.

CACHE

The current CACHE driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed doxygen issue.
- 2.0.3
  - Improvements
    - Deleted redundancy code about calculating cache clean/invalidate size and address aligns.
- 2.0.2
  - Bug Fixes
    - Fixed violation of MISRA C-2012 Rule 10.1, 10.3 and 10.4.
- 2.0.1
  - Bug Fixes
    - Fixed cache size issue in L2CACHE_GetDefaultConfig API.
- 2.0.0
  - Initial version.

COMMON

The current COMMON driver version is 2.4.1.

- 2.4.1
  - Improvements
    - Improve for the macro redefinition error when integrated with zephyr.
- 2.4.0
  - New Features
    - Added EnableIRQWithPriority, IRQ_SetPriority, and IRQ_ClearPendingIRQ for ARM.
    - Added MSDK_EnableCpuCycleCounter, MSDK_GetCpuCycleCount for ARM.
- 2.3.3
  - New Features
    - Added NETC into status group.
- 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_DELAY_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.

DCDC

The current DCDC driver version is 2.3.0.

- 2.3.0
  - Improvements
    * REG3[MISC_DELAY_TIMING], REG2[LOOPCTRL_DC_R], and REG2[LOOPCTRL_DC_C] are reserved in the latest RM, deleted corresponding functions.
- 2.2.1
  - Improvements
    * Fixed the doxygen warning.
- 2.2.0
  - New Features
    * Added supports for i.MXRT1170 series.
  - Bug Fixes
    * Fixed the warning that the DCDC_ConvertByteArrayToWord function defined but not used.
  - Improvements
    * Updated rcscale to reduce the ripple when booting into DCM.
- 2.1.0
  - Improvements
    * Divided the DCDC_AdjustTargetVoltage() into two APIs for two different modes.
  - Bug Fixes
    * Fixed the violations of MISRA C-2012 rules:
      · Rule 10.1, 10.4, 16.4, 17.7.
- 2.0.0
  - Initial version.

DMAMUX

The current DMAMUX driver version is 2.1.0.

- 2.1.0
  - Improvements
    * Modify the type of parameter source from uint32_t to int32_t in the DMAMUX_SetSource.
- 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.

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_GetRemainingMajorLoop-Count 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.

FLEXIO

The current FLEXIO driver version is 2.2.2.

- 2.2.2
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.2.1
  - Improvements
    - Added doxygen index parameter comment in FLEXIO_SetClockMode.
- 2.2.0
  - New Features
    - Added new APIs to support FlexIO pin register.
- 2.1.0
  - Improvements

        * Added API FLEXIO_SetClockMode to set flexio channel counter and source clock.
- 2.0.4
  - Bug Fixes
    * Fixed MISRA 8.4 issues.
- 2.0.3
  - Bug Fixes
    * Fixed MISRA 10.4 issues.
- 2.0.2
  - Improvements
    * Split FLEXIO component which combines all flexio/flexio_uart/flexio_i2c/flexio_i2s drivers into several components: FlexIO component, flexio_uart component, flexio_i2c_master component, and flexio_i2s component.
  - Bug Fixes
    * Fixed MISRA issues
      · Fixed rules 10.1, 10.3, 10.4, 10.7, 11.6, 11.9, 14.4, 17.7.
- 2.0.1
  - Bug Fixes
    * Fixed the dozen mode configuration error in FLEXIO_Init API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.

FLEXRAM

The current FLEXRAM driver version is 2.3.0.

- 2.3.0
  - New Features
    * Supported platforms which have ECC but no ECC error injection.
- 2.2.0
  - New Features
    * Supported flexram ECC error injection function.
- 2.1.0
  - New Features
    * Supported flexram ECC function.
- 2.0.7
  - Bug Fixes
    * Fixed doxygen issue.
- 2.0.6
  - New Features
    * Updated bank configuration and TCM size with GPR16/GPR17/GPR18 into SOC level for different SOC.
- 2.0.5
  - New Features
    * Added the magic address feature for OCRAM, DTCM and ITCM.
- 2.0.4

  - Bug Fixes
    - Fixed FlexRAM driver's missing extern C around functions in header file.
    - Removed magic address feature from driver.
- 2.0.3
  - Bug Fixes
    - Fixed the issue that TCM size configuration was wrong when TCM bank number was not a value power of 2.
- 2.0.2
  - Bug Fixes
    - Updated driver due to Reference Manual update.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA issue.
- 2.0.0
  - Initial version.

FLEXSPI

The current FLEXSPI driver version is 2.6.0.

- 2.6.0
  - New Features
    - Added new API to set AHB memory-mapped flash base address.
    - Added support of DLLxCR[REFPHASEGAP] bit field, it is recommended to set it as 0x2 if DLL calibration is enabled.
- 2.5.1
  - Bugfixes
    - Fixed handling of W1C bits in the INTR register
    - Removed FIFO resets from FLEXSPI_CheckAndClearError
    - FLEXSPI_TransferBlocking is observing IPCMDDONE and then fetches the final status of the transfer
    - Fixed issue that FLEXSPI2_DriverIRQHandler not defined.
- 2.5.0
  - Improvements
    - Supported word un-aligned access for write/read blocking/non-blocking API functions.
    - Fixed dead loop issue in DLL update function when using FRO clock source.
    - Fixed violations of the MISRA C-2012 Rule 10.3.
- 2.4.0
  - Improvements
    - Isolated IP command parallel mode and AHB command parallel mode using feature MACRO.
    - Supported new column address shift feature for external memory.
- 2.3.5
  - Bug Fixes

        * Fixed violations of the MISRA C-2012 Rule 14.2.
- 2.3.4
  - Bug Fixes
    * Updated flexspi_config_t structure and FlexSPI_Init to support new feature FSL_FEATURE_FLEXSPI_HAS_NO_MCR0_CONBINATION.
- 2.3.3
  - Bug Fixes
    * Removed feature FSL_FEATURE_FLEXSPI_DQS_DELAY_PS for DLL delay setting. Changed to use feature FSL_FEATURE_FLEXSPI_DQS_DELAY_MIN to set slave delay target as 0 for DLL enable and clock frequency higher than 100MHz.
- 2.3.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 Rule 8.4, 8.5, 10.1, 10.3, 10.4, 11.6 and 14.4.
- 2.3.1
  - Bug Fixes
    * Wait for bus to be idle before using it as access to external flash with new setting in FLEXSPI_SetFlashConfig() API.
    * Fixed the potential buffer overread and Tx FIFO overwrite issue in FLEXSPI_WriteBlocking.
- 2.3.0
  - New Features
    * Added new API FLEXSPI_UpdateDllValue for users to update DLL value after updating flexspi root clock.
    * Corrected grammatical issues for comments.
    * Added support for new feature FSL_FEATURE_FLEXSPI_DQS_DELAY_PS in DLL configuration.
- 2.2.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 Rule 10.1, 10.3 and 10.4.
    * Updated _flexspi_command from named enumerator into anonymous enumerator.
- 2.2.1
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.8, 11.9, 14.4, 15.7, 16.4, 17.7, 7.3.
    * Fixed IAR build warning Pe167.
    * Fixed the potential buffer overwrite and Rx FIFO overread issue in FLEXSPI_ReadBlocking.
- 2.2.0
  - Bug Fixes
    * Fixed flag name typos: kFLEXSPI_IpTxFifoWatermarkEmpltyFlag to kFLEXSPI_IpTxFifoWatermarkEmptyFlag; kFLEXSPI_IpCommandExcutionDoneFlag to kFLEXSPI_IpCommandExecutionDoneFlag.
    * Fixed comments typos such as sequencen->sequence, levle->level.
    * Fixed FLSHCR2[ARDSEQID] field clean issue.
    * Updated flexspi_config_t structure and FlexSPI_Init to support new feature FSL_FEAT-

URE_FLEXSPI_HAS_NO_MCR0_ATDFEN and FSL_FEATURE_FLEXSPI_HAS_NO_MCR0_ARDFEN.
    - Updated flexspi_flags_t structure to support new feature FSL_FEATURE_FLEXSPI_HAS_INTEN_AHBBUSERROREN.
- 2.1.1
  - Improvements
    - Defaulted enable prefetch for AHB RX buffer configuration in FLEXSPI_GetDefaultConfig, which is align with the reset value in AHBRXBUFxCR0.
    - Added software workaround for ERR011377 in FLEXSPI_SetFlashConfig; added some delay after DLL lock status set to ensure correct data read/write.
- 2.1.0
  - New Features
    - Added new API FLEXSPI_UpdateRxSampleClock for users to update read sample clock source after initialization.
    - Added reset peripheral operation in FLEXSPI_Init if required.
- 2.0.5
  - Bug Fixes
    - Fixed FLEXSPI_UpdateLUT cannot do partial update issue.
- 2.0.4
  - Bug Fixes
    - Reset flash size to zero for all ports in FLEXSPI_Init; fixed the possible out-of-range flash access with no error reported.
- 2.0.3
  - Bug Fixes
    - Fixed AHB receive buffer size configuration issue. The FLEXSPI_AHBRXBUFCR0_BUFSZ field should configure 64 bits size, and currently the AHB receive buffer size is in bytes which means 8-bit, so the correct configuration should be config->ahbConfig.buffer[i].bufferSize / 8.
- 2.0.2
  - New Features
    - Supported DQS write mask enable/disable feature during set FLEXSPI configuration.
    - Provided new API FLEXSPI_TransferUpdateSizeEDMA for users to update eDMA transfer size(SSIZE/DSIZE) per DMA transfer.
  - Bug Fixes
    - Fixed invalid operation of FLEXSPI_Init to enable AHB bus Read Access to IP RX FIFO.
    - Fixed incorrect operation of FLEXSPI_Init to configure IP TX FIFO watermark.
- 2.0.1
  - Bug Fixes
    - Fixed the flag clear issue and AHB read Command index configuration issue in FLEXSPI_SetFlashConfig.
    - Updated FLEXSPI_UpdateLUT function to update LUT table from any index instead of previous command index.
    - Added bus idle wait in FLEXSPI_SetFlashConfig and FLEXSPI_UpdateLUT to ensure bus is idle before any change to FlexSPI controller.
    - Updated interrupt API FLEXSPI_TransferNonBlocking and interrupt handle flow FLEX-

SPI_TransferHandleIRQ.
        - Updated eDMA API FLEXSPI_TransferEDMA.
- 2.0.0
  - Initial version.

GPC

The current GPC driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - Moved the assert sentence that irq register number has to be greater than 0 to platforms which irq 0-31 is not available.
    - Fixed the violations of MISRA C-2012 rules:
      - Rule 10.7 12.2.
- 2.1.0
  - Improvements
    - Updated driver for IMXRT.
- 2.0.0
  - Initial version.

GPT

The current GPT driver version is 2.0.5.

- 2.0.5
  - Improvements
    - Support workaround for ERR003777. This workaround helps switching the clock sources.
- 2.0.4
  - Bug Fixes
    - Fixed compiler warning when built with FSL_SDK_DISABLE_DRIVER_CLOCK_CONTROL flag enabled.
- 2.0.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 5.3 by customizing function parameter.
- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.8, 17.7.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.0.6.

- 2.0.6
  - Bug Fixes
    - Fixed compile warning: 'GPIO_GetInstance' defined but not used when macro FSL_SDK_DISABLE_DRIVER_CLOCK_CONTROL is defined.
- 2.0.5
  - Bug Fixes
    - Fixed MISRA C-2012 issue: rule-17.7.
- 2.0.4
  - Improvements
    - Updated the GPIO_PinWrite to use atomic operation if possible.
  - Bug Fixes
    - Fixed GPIO_PortToggle bug with platforms don't have register DR_TOGGLE.
- 2.0.3
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed rules, containing: rule-10.3, rule-14.4, and rule-15.5.
- 2.0.2
  - Bug Fixes
    - Fixed the bug of enabling wrong GPIO clock gate in initial API. Since some GPIO instances may not have a clock gate enabled, it checks the clock gate number and makes sure the clock gate is valid.
- 2.0.1
  - Improvements
    - API interface changes:
      - Refined naming of the API while keeping all original APIs, marking them as deprecated. Original APIs will be removed in next release. The main change is to update the API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

KPP

The current KPP driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules:
      - Rule 10.3 10.4 10.6 14.4 17.7
- 2.0.0
  - Initial version.

LPI2C

The current LPI2C driver version is 2.5.4.

- 2.5.4
  - Bug Fixes
    - Fixed LPI2C_MasterTransferBlocking() - the return value was sometime affected by call of LPI2C_MasterStop().
- 2.5.3
  - Improvements
    - Added handler for LPI2C7 and LPI2C8.
- 2.5.2
  - Bug Fixes
    - Fixed ERR051119 to ignore the nak flag when IGNACK=1 in LPI2C_MasterCheckAndClearError.
- 2.5.1
  - Bug Fixes
    - Added bus stop incase of bus stall in LPI2C_MasterTransferBlocking.
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.5.0
  - New Features
    - Added new function LPI2C_SlaveEnableAckStall to enable or disable ACKSTALL.
- 2.4.1
  - Improvements
    - Before master transfer with transactional APIs, enable master function while disable slave function and vise versa for slave transfer to avoid the one affecting the other.
- 2.4.0
  - Improvements
    - Split some functions, fixed CCM problem in file fsl_lpi2c.c.
  - Bug Fixes
    - Fixed bug in LPI2C_MasterInit that the MCFGR2's value set in LPI2C_MasterSetBaudRate may be overwritten by mistake.
- 2.3.2
  - Improvements
    - Initialized the EDMA configuration structure in the LPI2C EDMA driver.
- 2.3.1
  - Improvements
    - Updated LPI2C_GetCyclesForWidth to add the parameter of minimum cycle, because for master SDA/SCL filter, master bus idle/pin low timeout and slave SDA/SCL filter configuration, 0 means disabling the feature and cannot be used.
  - Bug Fixes
    - Fixed bug in LPI2C_SlaveTransferHandleIRQ that when restart detect event happens the transfer structure should not be cleared.
    - Fixed bug in LPI2C_RunTransferStateMachine, that when only slave address is transferred or there is still data remaining in tx FIFO the last byte's nack cannot be

ignored.
    - Fixed bug in slave filter doze enable, that when FILTDZ is set it means disable rather than enable.
    - Fixed bug in the usage of LPI2C_GetCyclesForWidth. First its return value cannot be used directly to configure the slave FILTSDA, FILTSCL, DATAVD or CLKHOLD, because the real cycle width for them should be FILTSDA+3, FILTSCL+3, FILTSCL+DATAVD+3 and CLKHOLD+3. Second when cycle period is not affected by the prescaler value, prescaler value should be passed as 0 rather than 1.
    - Fixed wrong default setting for LPI2C slave. If enabling the slave tx SCL stall, then the default clock hold time should be set to 250ns according to I2C spec for 100kHz standard mode baudrate.
    - Fixed bug that before pushing command to the tx FIFO the FIFO occupation should be checked first in case FIFO overflow.
- 2.3.0
  - New Features
    - Supported reading more than 256 bytes of data in one transfer as master.
    - Added API LPI2C_GetInstance.
  - Bug Fixes
    - Fixed bug in LPI2C_MasterTransferAbortEDMA, LPI2C_MasterTransferAbort and LPI2C_MasterTransferHandleIRQ that before sending stop signal whether master is active and whether stop signal has been sent should be checked, to make sure no FIFO error or bus error will be caused.
    - Fixed bug in LPI2C master EDMA transactional layer that the bus error cannot be caught and returned by user callback, by monitoring bus error events in interrupt handler.
    - Fixed bug in LPI2C_GetCyclesForWidth that the parameter used to calculate clock cycle should be 2^prescaler rather than prescaler.
    - Fixed bug in LPI2C_MasterInit that timeout value should be configured after baudrate, since the timeout calculation needs prescaler as parameter which is changed during baudrate configuration.
    - Fixed bug in LPI2C_MasterTransferHandleIRQ and LPI2C_RunTransferStateMachine that when master writes with no stop signal, need to first make sure no data remains in the tx FIFO before finishes the transfer.
- 2.2.0
  - Bug Fixes
    - Fixed issue that the SCL high time, start hold time and stop setup time do not meet I2C specification, by changing the configuration of data valid delay, setup hold delay, clock high and low parameters.
    - MISRA C-2012 issue fixed.
      - Fixed rule 8.4, 13.5, 17.7, 20.8.
- 2.1.12
  - Bug Fixes
    - Fixed MISRA advisory 15.5 issues.
- 2.1.11
  - Bug Fixes
    - Fixed the bug that, during master non-blocking transfer, after the last byte is sent/received,

the kLPI2C_MasterNackDetectFlag is expected, so master should not check and clear kLPI2C_MasterNackDetectFlag when remainingBytes is zero, in case FIFO is emptied when stop command has not been sent yet.
    - Fixed the bug that, during non-blocking transfer slave may nack master while master is busy filling tx FIFO, and NDF may not be handled properly.
- 2.1.10
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed rule 10.3, 14.4, 15.5.
    - Fixed unaligned access issue in LPI2C_RunTransferStateMachine.
    - Fixed uninitialized variable issue in LPI2C_MasterTransferHandleIRQ.
    - Used linked TCD to disable tx and enable rx in read operation to fix the issue that for platform sharing the same DMA request with tx and rx, during LPI2C read operation if interrupt with higher priority happened exactly after command was sent and before tx disabled, potentially both tx and rx could trigger dma and cause trouble.
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 11.6, 11.9, 14.4, 17.7.
    - Fixed the waitTimes variable not re-assignment issue for each byte read.
  - New Features
    - Added the IRQHandler for LPI2C5 and LPI2C6 instances.
  - Improvements
    - Updated the LPI2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
- 2.1.9
  - Bug Fixes
    - Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
    - Fixed Coverity issue of operands did not affect the result in LPI2C_SlaveReceive and LPI2C_SlaveSend.
    - Removed STOP signal wait when NAK detected.
    - Cleared slave repeat start flag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved repeat start flag. This caused the next slave to send a break, and the master was always in the receive data status, but could not receive data.
- 2.1.8
  - Bug Fixes
    - Fixed the transfer issue with LPI2C_MasterTransferNonBlocking, kLPI2C_TransferNoStopFlag, with the wait transfer done through callback in a way of not doing a blocking transfer.
    - Fixed the issue that STOP signal did not appear in the bus when NAK event occurred.
- 2.1.7
  - Bug Fixes
    - Cleared the stopflag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved stop flag and caused the next slave to send a break, and the master always stayed in the receive data status but could not receive data.
- 2.1.6

  - Bug Fixes
    * Fixed driver MISRA build error and C++ build error in LPI2C_MasterSend and LPI2C_-SlaveSend.
    * Reset FIFO in LPI2C Master Transfer functions to avoid any byte still remaining in FIFO during last transfer.
    * Fixed the issue that LPI2C_MasterStop did not return the correct NAK status in the bus for second transfer to the non-existing slave address.
- 2.1.5
  - Bug Fixes
    * Extended the Driver IRQ handler to support LPI2C4.
    * Changed to use ARRAY_SIZE(kLpi2cBases) instead of FEATURE COUNT to decide the array size for handle pointer array.
- 2.1.4
  - Bug Fixes
    * Fixed the LPI2C_MasterTransferEDMA receive issue when LPI2C shared same request source with TX/RX DMA request. Previously, the API used scatter-gather method, which handled the command transfer first, then the linked TCD which was pre-set with the receive data transfer. The issue was that the TX DMA request and the RX DMA request were both enabled, so when the DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happened due to empty TX FIF-O. The result was that the RX DMA transfer would start without waiting on the expected RX DMA request.
    * Fixed the issue by enabling IntMajor interrupt for the command TCD and checking if there was a linked TCD to disable the TX DMA request in LPI2C_MasterEDMACallback API.
- 2.1.3
  - Improvements
    * Added LPI2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
    * Added LPI2C_MasterTransferBlocking API.
- 2.1.2
  - Bug Fixes
    * In LPI2C_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag was detected.
- 2.1.1
  - Bug Fixes
    * Disabled the auto-stop feature in eDMA driver. Previously, the auto-stop feature was enabled at transfer when transferring with stop flag. Since transfer was without stop flag and the auto-stop feature was enabled, when starting a new transfer with stop flag, the stop flag would be sent before the new transfer started, causing unsuccesful sending of the start flag, so the transfer could not start.
    * Changed default slave configuration with address stall false.
- 2.1.0
  - Improvements
    * API name changed:

  - LPI2C_MasterTransferCreateHandle -> LPI2C_MasterCreateHandle.
  - LPI2C_MasterTransferGetCount -> LPI2C_MasterGetTransferCount.
  - LPI2C_MasterTransferAbort -> LPI2C_MasterAbortTransfer.
  - LPI2C_MasterTransferHandleIRQ -> LPI2C_MasterHandleInterrupt.
  - LPI2C_SlaveTransferCreateHandle -> LPI2C_SlaveCreateHandle.
  - LPI2C_SlaveTransferGetCount -> LPI2C_SlaveGetTransferCount.
  - LPI2C_SlaveTransferAbort -> LPI2C_SlaveAbortTransfer.
  - LPI2C_SlaveTransferHandleIRQ -> LPI2C_SlaveHandleInterrupt.
- 2.0.0
  - Initial version.

LPSPI

The current LPSPI driver version is 2.6.8.

- 2.6.8
  - Bug Fixes
    - Fixed build error when SPI_RETRY_TIMES is defined to non-zero value.
- 2.6.7
  - Bug Fixes
    - Fixed the txData from void * to const void * in transmit API _lpspi_master_handle and _lpspi_slave_handle.
- 2.6.6
  - Bug Fixes
    - Added LPSPI register init in LPSPI_MasterInit incase of LPSPI register exist.
- 2.6.5
  - Improvements
    - Introduced FSL_FEATURE_LPSPI_HAS_NO_PCSCFG and FSL_FEATURE_LPSPI_-HAS_NO_MULTI_WIDTH for conditional compile.
    - Release peripheral from reset if necessary in init function.
- 2.6.4
  - Bug Fixes
    - Added LPSPI6_DriverIRQHandler for LPSPI6 instance.
- 2.6.3
  - Hot Fixes
    - Added macro switch in function LPSPI_Enable about ERRATA051472.
- 2.6.2
  - Bug Fixes
    - Disabled lpspi before LPSPI_MasterSetBaudRate incase of LPSPI opened.
- 2.6.1
  - Bug Fixes
    - Fixed return value while calling LPSPI_WaitTxFifoEmpty in function LPSPI_Master-TransferNonBlocking.
- 2.6.0

  - Feature
    - Added the new feature of multi-IO SPI .
- 2.5.3
  - Bug Fixes
    - Fixed 3-wire txmask of handle vaule reentrant issue.
- 2.5.2
  - Bug Fixes
    - Workaround for errata ERR051588 by clearing FIFO after transmit underrun occurs.
- 2.5.1
  - Bug Fixes
    - Workaround for errata ERR050456 by resetting the entire module using LPSPIn_CR[RST] bit.
- 2.5.0
  - Bug Fixes
    - Workaround for errata ERR011097 to wait the TX FIFO to go empty when writing TCR register and TCR[TXMSK] value is 1.
    - Added API LPSPI_WaitTxFifoEmpty for wait the txfifo to go empty.
- 2.4.7
  - Bug Fixes
    - Fixed bug that the SR[REF] would assert if software disabled or enabled the LPSPI module in LPSPI_Enable.
- 2.4.6
  - Improvements
    - Moved the configuration of registers for the 3-wire lpspi mode to the LPSPI_MasterInit and LPSPI_SlaveInit function.
- 2.4.5
  - Improvements
    - Improved LPSPI_MasterTransferBlocking send performance when frame size is 1-byte.
- 2.4.4
  - Bug Fixes
    - Fixed LPSPI_MasterGetDefaultConfig incorrect default inter-transfer delay calculation.
- 2.4.3
  - Bug Fixes
    - Fixed bug that the ISR response speed is too slow on some platforms, resulting in the first transmission of overflow, Set proper RX watermarks to reduce the ISR response times.
- 2.4.2
  - Bug Fixes
    - Fixed bug that LPSPI_MasterTransferBlocking will modify the parameter txbuff and rxbuff pointer.
- 2.4.1
  - Bug Fixes
    - Fixed bug that LPSPI_SlaveTransferNonBlocking can't detect RX error.
- 2.4.0
  - Improvements
    - Split some functions, fixed CCM problem in file fsl_lpspi.c.

- 2.3.1
  - Improvements
    - Initialized the EDMA configuration structure in the LPSPI EDMA driver.
  - Bug Fixes
    - Fixed bug that function LPSPI_MasterTransferBlocking should return after the transfer complete flag is set to make sure the PCS is re-asserted.
- 2.3.0
  - New Features
    - Supported the master configuration of sampling the input data using a delayed clock to improve slave setup time.
- 2.2.1
  - Bug Fixes
    - Fixed bug in LPSPI_SetPCSContinous when disabling PCS continous mode.
- 2.2.0
  - Bug Fixes
    - Fixed bug in 3-wire polling and interrupt transfer that the received data is not correct and the PCS continous mode is not working.
- 2.1.0
  - Improvements
    - Improved LPSPI_SlaveTransferHandleIRQ to fill up TX FIFO instead of write one data to TX register which improves the slave transmit performance.
    - Added new functional APIs LPSPI_SelectTransferPCS and LPSPI_SetPCSContinous to support changing PCS selection and PCS continous mode.
  - Bug Fixes
    - Fixed bug in non-blocking and EDMA transfer APIs that kStatus_InvalidArgument is returned if user configures 3-wire mode and full-duplex transfer at the same time, but transfer state is already set to kLPSPI_Busy by mistake causing following transfer can not start.
    - Fixed bug when LPSPI slave using EDMA way to transfer, tx should be masked when tx data is null, otherwise in 3-wire mode which tx/rx use the same pin, the received data will be interfered.
- 2.0.5
  - Improvements
    - Added timeout mechanism when waiting certain states in transfer driver.
  - Bug Fixes
    - Fixed the bug that LPSPI can not transfer large data using EDMA.
    - Fixed MISRA 17.7 issues.
    - Fixed variable overflow issue introduced by MISRA fix.
    - Fixed issue that rxFifoMaxBytes should be calculated according to transfer width rather than FIFO width.
    - Fixed issue that completion flag was not cleared after transfer completed.
- 2.0.4
  - Bug Fixes
    - Fixed in LPSPI_MasterTransferBlocking that master rxfifo may overflow in stall condition.

        * Eliminated IAR Pa082 warnings.
        * Fixed MISRA issues.
            · Fixed rules 10.1, 10.3, 10.4, 10.6, 11.9, 14.2, 14.4, 15.7, 17.7.
- 2.0.3
  - Bug Fixes
    * Removed LPSPI_Reset from LPSPI_MasterInit and LPSPI_SlaveInit, because this API may glitch the slave select line. If needed, call this function manually.
- 2.0.2
  - New Features
    * Added dummy data set up API to allow users to configure the dummy data to be transferred.
    * Enabled the 3-wire mode, SIN and SOUT pins can be configured as input/output pin.
- 2.0.1
  - Bug Fixes
    * Fixed the bug that the clock source should be divided by the PRESCALE setting in LPSPI_MasterSetDelayTimes function.
    * Fixed the bug that LPSPI_MasterTransferBlocking function would hang in some corner cases.
  - Optimization
    * Added #ifndef/#endif to allow user to change the default TX value at compile time.
- 2.0.0
  - Initial version.

LPSPI_EDMA

The current LPSPI_EDMA driver version is 2.4.4.

- 2.4.4
  - Improvements
    * Add EDMA ext API to accommodate more types of EDMA.
- 2.4.3
  - Improvements
    * Supported 32K bytes transmit in DMA, improve the max datasize in LPSPI_MasterTransferEDMALite.
- 2.4.2
  - Improvements
    * Added callback status in EDMA_LpspiMasterCallback and EDMA_LpspiSlaveCallback to check transferDone.
- 2.4.1
  - Improvements
    * Add the TXMSK wait after TCR setting.
- 2.4.0
  - Improvements
    * Separated LPSPI_MasterTransferEDMA functions to LPSPI_MasterTransferPrepareED-

MA and LPSPI_MasterTransferEDMALite to optimize the process of transfer.

LPUART

The current LPUART driver version is 2.8.2.

- 2.8.2
  - Bug Fix
    - Fixed the bug that LPUART_TransferEnable16Bit controled by wrong feature macro.
- 2.8.1
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule-5.3, rule-5.8, rule-10.4, rule-11.3, rule-11.8.
- 2.8.0
  - Improvements
    - Added support of DATA register for 9bit or 10bit data transmit in write and read API. Such as: LPUART_WriteBlocking16bit, LPUART_ReadBlocking16bit, LPUART_TransferEnable16Bit LPUART_WriteNonBlocking16bit, LPUART_ReadNonBlocking16bit.
- 2.7.7
  - Bug Fixes
    - Fixed the bug that baud rate calculation overflow when srcClock_Hz is 528MHz.
- 2.7.6
  - Bug Fixes
    - Fixed LPUART_EnableInterrupts and LPUART_DisableInterrupts bug that blocks if the LPUART address doesn't support exclusive access.
- 2.7.5
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.7.4
  - Improvements
    - Added support for atomic register accessing in LPUART_EnableInterrupts and LPUART_DisableInterrupts.
- 2.7.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 15.7.
- 2.7.2
  - Bug Fix
    - Fixed the bug that the OSR calculation error when lupart init and lpuart set baud rate.
- 2.7.1
  - Improvements
    - Added support for LPUART_BASE_PTRS_NS in security mode in file fsl_lpuart.c.
- 2.7.0
  - Improvements
    - Split some functions, fixed CCM problem in file fsl_lpuart.c.

- 2.6.0
  - Bug Fixes
    - Fixed bug that when there are multiple lpuart instance, unable to support different ISR.
- 2.5.3
  - Bug Fixes
    - Fixed comments by replacing unused status flags kLPUART_NoiseErrorInRxDataRegFlag 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_TransferEdmaHandleIRQ 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/eDM-

A 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 LPUART_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

OCOTP

The current OCOTP driver version is 2.1.3.

- 2.1.3
  - Bug fixes
    - Fixed MISRA 2012 issue: 8.4, 10.3, 10.4, 14.3.
    - Fixed doxygen warning.
- 2.1.2
  - Improvements
    - Updated for new MIMXRT117X header file.
- 2.1.1
  - Improvements
    - Updated OCOTP_ReloadShadowRegister to return error status.
    - Added functions OCOTP_ReadFuseShadowRegisterExt and OCOTP_WriteFuseShadowRegisterWithLock.
  - Bug fixes
    - Fixed MISRA 2012 rule 10.3 issue.
- 2.0.1
  - Bug Fixes
    - Fixed doxygen issues.
- 2.0.0
  - Initial version.

OTFAD

The current driver version is 2.1.4.

- 2.1.4
  - Bug fixes
    - Fixed MISRA 2012 issue: 10.1.

- 2.1.3
  - Bug fixes
    - Fixed the error that waiting for both FLEXSPI AHB idle and SEQ idle.
- 2.1.2
  - Bug fixes
    - Fixed MISRA 2012 issue: 10.4.
- 2.1.1
  - Improvements:
    - Hided some bits in CR and SR registers for selected platforms.
    - Fixed doxygen issues.
- 2.1.0
  - Improvements:
    - Used boolean type to define 1-bit field concepts.
- 2.0.0
  - Initial version.

PIT

The current PIT driver version is 2.0.5.

- 2.0.5
  - Improvements
    - Support workaround for ERR007914. This workaround guarantee the write to MCR register is not ignored.
- 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.

PMU

The current PMU driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules: Rule 10.1 10.4
- 2.1.0
  - Improvements
    - Added feature macros for low power control APIs to support conditional compile.
    - Renamed "PMU_2P1EnablePullDown" to "PMU_2P5EnablePullDown".
- 2.0.0
  - Initial version.

PWM

The current PWM driver version is 2.8.4.

- 2.8.4
  - Improvements
    - Support workaround for ERR051989. This function helps realize no phase delay between submodule 0 and other submodule.
- 2.8.3
  - Bug Fixes
    - Fixed MISRA C-2012 Rule 15.7
- 2.8.2
  - Bug Fixes
    - Fixed warning conversion from ’int’ to ’uint16_t’ on API PWM_Init.
    - Fixed warning unused variable ’reg’ on API PWM_SetPwmForceOutputToZero.
- 2.8.1
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.8.0
  - Improvements
    - Added API PWM_UpdatePwmPeriodAndDutycycle to update the PWM signal’s period and dutycycle for a PWM submodule.
    - Added API PWM_SetPeriodRegister and PWM_SetDutycycleRegister to merge duplicate code in API PWM_SetupPwm, PWM_UpdatePwmDutycycleHighAccuracy and PWM_-UpdatePwmPeriodAndDutycycle
- 2.7.1
  - Improvements
    - Supported UPDATE_MASK bit in MASK register.
- 2.7.0
  - Improvements
    - Supported platforms which don’t have Capture feature with channel A and B.

    - Supported platforms which don't have Submodule 3.
    - Added assert function in API PWM_SetPhaseDelay to prevent wrong argument.
- 2.6.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 10.3.
- 2.6.0
  - Improvements
    - Added API PWM_SetPhaseDelay to set the phase delay from the master sync signal of submodule 0.
    - Added API PWM_SetFilterSampleCountthe to set number of consecutive samples that must agree prior to the input filter.
    - Added API PWM_SetFilterSamplePeriod to set set the sampling period of the fault pin input filter.
- 2.5.1
  - Bug Fixes
    - Fixed MISRA C-2012 rules: 10.1, 10.3, 10.4 , 10.6 and 10.8.
    - Fixed the issue that PWM_UpdatePwmDutycycle() can't update duty cycle status value correct.
- 2.5.0
  - Improvements
    - Added API PWM_SetOouputToIdle to set pwm channel output to idle.
    - Added API PWM_GetPwmChannelState to get the pwm channel output duty cycle value.
    - Added API PWM_SetPwmForceOutputToZero to set the pwm channel output to zero logic.
    - Added API PWM_SetChannelOutput to set the pwm channel output state.
    - Added API PWM_SetClockMode to set the value of the clock prescaler.
    - Added API PWM_SetupPwmPhaseShift to set PWM which a special phase shift and 50% duty cycle.
    - Added API PWM_SetVALxValue/PWM_GetVALxValue to set/get PWM VALs registers values directly.
- 2.4.0
  - Improvements
    - Supported the PWM which can't work in wait mode.
- 2.3.0
  - Improvements
    - Add PWM output enable&disbale API for SDK.
  - Bug Fixes
    - Fixed changing channel B configuration when parameter is kPWM_PWMX and PWMX configuration is not supported yet.
- 2.2.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 10.3, 10.4.
  - Bug Fixes
    - Fixed the issue that PWM drivers computed VAL1 improperly.
  - Improvements

        * Updated calculation accuracy of reloadValue in dutyCycleToReloadValue function.
- 2.2.0
  - Improvements
    * Added new enumeration and two APIs to support enabling and disabling one or more PWM output triggers.
    * Added a new function to make the most of 16-bit resolution PWM.
    * Added one API to support updating fault status of PWM output.
    * Added one API to support PWM DMA write request.
    * Added three APIs to support PWM DMA capture read request.
    * Added one API to support get default fault config of PWM.
    * Added one API to support setting PWM fault disable mapping.
- 2.1.0
  - Improvements
    * Moved the configuration of fault input filter into a new API to avoid be initialized multiple times.
  - Bug Fixes
    * MISRA C-2012 issue fixed.
      · Fix rules, containing: rule-10.2, rule-10.3, rule-10.4, rule-10.7, rule-10.8, rule-14.4, rule-16.4.
- 2.0.1
  - Bug Fixes
    * Fixed the issue that PWM submodule may be initialized twice in function PWM_SetupPwm().
- 2.0.0
  - Initial version.

RTWDOG

The current RTWDOG driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    * Fixed doxygen issue.
- 2.1.1
  - Bug Fixes
    * MISRA C-2012 issue fixed.
      · Fixed rules, containing: rule-10.3, rule-10.8, rule-11.9, rule-14.4, rule-15.5.
- 2.1.0
  - Improvements
    * Added an API to enable or disable the window mode.
    * Added an API to convert a raw count value to millisecond.
    * Used AT_QUICKACCESS_SECTION_CODE macro to decorate RTWDOG_Init, and copied this function from flash to QUICKACCESS section.
- 2.0.1

  - Bug Fixes
    - Fixed bug in the RTWDOG_Init; added check for register's unlock status when configuring the RTWDOG in RTWDOG_init.
- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.4.2

- 2.4.2
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.4.1
  - Bug Fixes
    - Fixed bitWidth incorrectly assigned issue.
- 2.4.0
  - Improvements
    - Removed deprecated APIs.
- 2.3.8
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.4.
- 2.3.7
  - Improvements
    - Change feature "FSL_FEATURE_SAI_FIFO_COUNT" to "FSL_FEATURE_SAI_HAS_FIFO".
    - Added feature "FSL_FEATURE_SAI_FIFO_COUNTn(x)" to align SAI fifo count function with IP in function
- 2.3.6
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 5.6.
- 2.3.5
  - Improvements
    - Make driver to be aarch64 compatible.
- 2.3.4
  - Bug Fixes
    - Corrected the fifo combine feature macro used in driver.
- 2.3.3
  - Bug Fixes
    - Added bit clock polarity configuration when sai act as slave.
    - Fixed out of bound access coverity issue.
    - Fixed violations of MISRA C-2012 rule 10.3, 10.4.
- 2.3.2
  - Bug Fixes

        * Corrected the frame sync configuration when sai act as slave.
- 2.3.1
  - Bug Fixes
    * Corrected the peripheral name in function SAI0_DriverIRQHandler.
    * Fixed violations of MISRA C-2012 rule 17.7.
- 2.3.0
  - Bug Fixes
    * Fixed the build error caused by the SOC has no fifo feature.
- 2.2.3
  - Bug Fixes
    * Corrected the peripheral name in function SAI0_DriverIRQHandler.
- 2.2.2
  - Bug Fixes
    * Fixed the issue of MISRA 2004 rule 9.3.
    * Fixed sign-compare warning.
    * Fixed the PA082 build warning.
    * Fixed sign-compare warning.
    * Fixed violations of MISRA C-2012 rule 10.3,17.7,10.4,8.4,10.7,10.8,14.4,17.7,11.6,10.-1,10.6,8.4,14.3,16.4,18.4.
    * Allow to reset Rx or Tx FIFO pointers only when Rx or Tx is disabled.
  - Improvements
    * Added 24bit raw audio data width support in sai sdma driver.
    * Disabled the interrupt/DMA request in the SAI_Init to avoid generates unexpected sai FIFO requests.
- 2.2.1
  - Improvements
    * Added mclk post divider support in function SAI_SetMasterClockDivider.
    * Removed useless configuration code in SAI_RxSetSerialDataConfig.
  - Bug Fixes
    * Fixed the SAI SDMA driver build issue caused by the wrong structure member name used in the function SAI_TransferRxSetConfigSDMA/SAI_TransferTxSetConfigSDMA.
    * Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CHANNEL_COUNTn.
    * Applied ERR05144: not set FCONT = 1 when TMR > 0, otherwise the TX may not work.
- 2.2.0
  - Improvements
    * Added new APIs for parameters collection and simplified user interfaces:
      · SAI_Init
      · SAI_SetMasterClockConfig
      · SAI_TxSetBitClockRate
      · SAI_TxSetSerialDataConfig
      · SAI_TxSetFrameSyncConfig
      · SAI_TxSetFifoConfig
      · SAI_TxSetBitclockConfig
      · SAI_TxSetConfig

        - SAI_TxSetTransferConfig
        - SAI_RxSetBitClockRate
        - SAI_RxSetSerialDataConfig
        - SAI_RxSetFrameSyncConfig
        - SAI_RxSetFifoConfig
        - SAI_RxSetBitclockConfig
        - SAI_RXSetConfig
        - SAI_RxSetTransferConfig
        - SAI_GetClassicI2SConfig
        - SAI_GetLeftJustifiedConfig
        - SAI_GetRightJustifiedConfig
        - SAI_GetTDMConfig
- 2.1.9
  - Improvements
    - Improved SAI driver comment for clock polarity.
    - Added enumeration for SAI for sample inputs on different edges.
    - Changed FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHANNEL_COUNTn(base) for the difference between the different SAI instances.
  - Added new APIs:
    - SAI_TxSetBitClockDirection
    - SAI_RxSetBitClockDirection
    - SAI_RxSetFrameSyncDirection
    - SAI_TxSetFrameSyncDirection
- 2.1.8
  - Improvements
    - Added feature macro test for the sync mode2 and mode 3.
    - Added feature macro test for masterClockHz in sai_transfer_format_t.
- 2.1.7
  - Improvements
    - Added feature macro test for the mclkSource member in sai_config_t.
    - Changed "FSL_FEATURE_SAI5_SAI6_SHARE_IRQ" to "FSL_FEATURE_SAI_SAI5_SAI6_SHARE_IRQ".
    - Added #ifndef #endif check for SAI_XFER_QUEUE_SIZE to allow redefinition.
  - Bug Fixes
    - Fixed build error caused by feature macro test for mclkSource.
- 2.1.6
  - Improvements
    - Added feature macro test for mclkSourceClockHz check.
    - Added bit clock source name for general devices.
  - Bug Fixes
    - Fixed incorrect channel numbers setting while calling RX/TX set format together.
- 2.1.5
  - Bug Fixes
    - Corrected SAI3 driver IRQ handler name.
    - Added I2S4/5/6 IRQ handler.

    - Added base in handler structure to support different instances sharing one IRQ number.
  - New Features
    - Updated SAI driver for MCR bit MICS.
    - Added 192 KHZ/384 KHZ in the sample rate enumeration.
    - Added multi FIFO interrupt/SDMA transfer support for TX/RX.
    - Added an API to read/write multi FIFO data in a blocking method.
    - Added bclk bypass support when bclk is same with mclk.
- 2.1.4
  - New Features
    - Added an API to enable/disable auto FIFO error recovery in platforms that support this feature.
    - Added an API to set data packing feature in platforms which support this feature.
- 2.1.3
  - New Features
    - Added feature to make I2S frame sync length configurable according to bitWidth.
- 2.1.2
  - Bug Fixes
    - Added 24-bit support for SAI eDMA transfer. All data shall be 32 bits for send/receive, as eDMA cannot directly handle 3-Byte transfer.
- 2.1.1
  - Improvements
    - Reduced code size while not using transactional API.
- 2.1.0
  - Improvements
    - API name changes:
      - SAI_GetSendRemainingBytes -> SAI_GetSentCount.
      - SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount.
      - All names of transactional APIs were added with "Transfer" prefix.
      - All transactional APIs use base and handle as input parameter.
      - Unified the parameter names.
  - Bug Fixes
    - Fixed WLC bug while reading TCSR/RCSR registers.
    - Fixed MOE enable flow issue. Moved MOE enable after MICS settings in SAI_TxInit/SAI_RxInit.
- 2.0.0
  - Initial version.

SPDIF

The current SPDIF driver version is 2.0.7.

- 2.0.7
  - Improvements
    - Add feature macro FSL_FEATURE_SPDIF_HAS_NO_SIC_REGISTER to handle

nonexistent SIC register.

- 2.0.6
  - Bug Fixes
    - Fixed the Q/U channel interrupt enabled unexpectly while Q/U transfer pointer is NULL.
- 2.0.5
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.3.
- 2.0.4
  - Bug Fixes
    - Added udata/qdata buffer address validation in driver IRQ handler to ensure that NULL pointer dereferences do not occur.
- 2.0.3
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3, 10.4, and 14.4.
- 2.0.2
  - Bug Fixes
    - Corrected operator used for size value assertion in SPDIF_ReadBlocking/SPDIF_WriteBlocking.
- 2.0.1
  - Bug Fixes
    - Corrected the feature macro name used to define s_edmaPrivateHandle.
- 2.0.0
  - Initial version.

SRC

The current SRC driver version is 2.0.1.

- 2.0.1
  - Improvements
    - Updated SRC driver for adding SRC_SRSR_JTAG_SW_RST enumeration.
- 2.0.0
  - Initial version.

TEMPMON

The current TEMPMON driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules:
      - Rule 10.3 10.4.
- 2.1.0
  - Bug Fixes

        * Supported minus value for alarm temperature setting.
        * Fixed wrong temperature calculation equation.
- 2.0.3
  - Improvements
    * Added temperature threshold check for high/low/panic to avoid temperature overflow.
- 2.0.2
  - Bug Fixes
    * Fixed wrong alarm value setting API, it need to clear it firstly and set a new value into it.
- 2.0.1
  - Bug Fixes
    * Fixed the violations of MISRA C-2012 rules:
      · Rule 10.1 10.3 10.4 10.8 17.7.
- 2.0.0
  - Initial version.

WDOG

The current WDOG driver version is 2.2.0.

- 2.2.0
  - Bug Fixes
    * Fixed the wrong behavior of workMode.enableWait, workMode.enableStop, workMode.-enableDebug in configuration structure wdog_config_t. When set the items to true, WDOG will continues working in those modes.
- 2.1.1
  - Bug Fixes
    * MISRA C-2012 issue fixed: rule 10.1, 10.3, 10.4, 10.6, 10.7 and 11.9.
    * Fixed the issue of the inseparable process interrupted by other interrupt source.
      · WDOG_Init
      · WDOG_Refresh
- 2.1.0
  - New Features
    * Added new API "WDOG_TriggerSystemSoftwareReset()" to allow users to reset the system by software.
    * Added new API "WDOG_TriggerSoftwareSignal()" to allow users to trigger a WDOG_B signal by software.
    * Removed the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog-_config_t structure.
    * Added new parameter "enableTimeOutAssert" to the wdog_config_t structure. With this parameter enabled, when the WDOG timeout occurs, a WDOG_B signal will be asserted. This signal can be routed to external pin of the chip. Note that WDOG_B signal remains asserted until a power-on reset (POR) occurs.
- 2.0.1
  - New Features

    - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

XBARA

The current XBARA driver version is 2.0.6.

- 2.0.6
  - Bug Fixes
    - Fixed typo in kXBARA_RequestInterruptEnalbe item.
- 2.0.5
  - Bug Fixes
    - Fixed IAR build warning Pa082.
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.7, 10.8, 12.1, 18.1, 20.7.
- 2.0.4
  - Improvements
    - Optimized XBARA_SetOutputSignalConfig.
- 2.0.3
  - Bug Fixes
    - Corrected configuration for function XBAR_SetOutputSignalConfig.
- 2.0.2
  - Other Changes
    - Changed array clock name.
- 2.0.1
  - Bug Fixes
    - Fixed w1c bits for XBARA_SetOutputSignalConfig function.
- 2.0.0
  - Initial version.

2 Middleware Change Log

emWin library

The currently supported version is 6.38

- v6.38
  - upgraded to v6.38
- v6.34c
  - upgraded to v6.34c
- v6.28_rev1
  - add cm33_nodsp_fpu libraries for Cortec M33 without DSP extension with SP FPU
- v6.28
  - upgraded to v6.28
- v6.24_rev2
  - add cm33_nodsp libraries for Cortex M33 without DSP extension
- v6.24_rev1
  - recompiled cm33 library with fpu single precision
  - added cm7_sp library for Cortex M7 with sp fpu for IAR
- v6.24
  - upgraded to v6.24
- v6.16c
  - upgraded to v6.16c
  - updated temperature_control demo generated by AppWizard
- v6.14d
  - upgraded to v6.14d
- v6.10f
  - upgraded to v6.10f

FatFs for MCUXpresso SDK

Current version is FatFs R0.15_rev0.

- R0.15_rev0
  - Upgraded to version 0.15
  - Applied patches from http://elm-chan.org/fsw/ff/patches.html
- R0.14b_rev1
  - Applied patches from http://elm-chan.org/fsw/ff/patches.html
- 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 3.0.6. 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
- 3.0.6
  - General update for SDK 2.13
  - Use of new Ethernet MDIO driver concept.
  - Support of ENET and NETC Ethernet modules in the fmstr_net example application.
- 3.0.7
  - General update for SDK 2.14

LVGL for KSDK

- 8.3.10_rev1
  - Integrate LVGL 8.3.10 to SDK.
- 8.3.9_rev1
  - Integrate LVGL 8.3.9 to SDK.
- 8.3.5_rev1
  - Integrate LVGL 8.3.5 to SDK.
- 8.3.2_rev1
  - Integrate LVGL 8.3.2 to SDK.
- 8.3.0_rev1
  - Integrate LVGL 8.3.0 to SDK.
- 8.2.0_rev1
  - Integrate LVGL 8.2.0 to SDK.
- 8.0.2_rev1
  - Integrate LVGL 8.0.2 to SDK.
- 7.10.1_rev1

  - Integrate LVGL 7.10.1 to SDK.
  - Added PXP, VGLite hardware acceleration.
- 7.4.0_rev1
  - Integrate LVGL 7.4.0 to SDK.
- 7.0.0_rev1
  - Integrate LVGL 7.0.0 to SDK.
  - Added PXP hardware acceleration initial version.
- 6.1.1_rev1
  - Integrate LVGL 6.1.1 to SDK.
- 5.3_rev1
  - Integrate LVGL 5.3 to SDK.

mbedTLS for MCUXpresso SDK

The current version of mbedTLS is based on mbed TLS 2.28.8 branch released 2024-03-28

- 2.28.8
  - New features:
    - Ported mbedTLS 2.28.8 to SDK.
- 2.28.5
  - New features:
    - Ported mbedTLS 2.28.5 to SDK.
- 2.28.4
  - New features:
    - Ported mbedTLS 2.28.4 to SDK.
- 2.28.3
  - New features:
    - Ported mbedTLS 2.28.3 to SDK.
- 2.28.1
  - New features:
    - Ported mbedTLS 2.28.1 to SDK.
- 2.28.0
  - New features:
    - Ported mbedTLS 2.28.0 to SDK.
- 2.27.0
  - New features:
    - Ported mbedTLS 2.27.0 to SDK.
- 2.26.0
  - New features:
    - Ported mbedTLS 2.26.0 to SDK.
- 2.16.6_rev7
  - Bug fixes:
    - Corrected definition of global variable g_isCryptoHWInitialized to be only internal static variable in sssapi_mbedtls.c file.

- 2.16.6_rev6
  - Bug fixes:
    - Adding #ifdef in ecdsa.c to remove warning: "function "derive_mpi" was declared but never referenced", when alternative implementation of ECDSA sign and verify is used and not used Deterministic ECDSA, then was derive_mpi function never used.
- 2.16.6_rev5
  - New features:
    - Changed return type of CRYPTO_InitHardware() from void to status_t. Added check of this return value in selftest.c and benchmark.c files.
- 2.16.6_rev4
  - New features:
    - Added mutex for HW modules HASHCRYPT and CASPER. Enabled by MBEDTLS_THREADING_C
- 2.16.6_rev3
  - New features:
    - Added support for KW45 device with latest Sentinel200. Port of SSS API mbedtls implementation to KW45.
- 2.16.6_rev2
  - New features:
    - Added support for SW computing AES-192/256 while using DCP driver.
- 2.16.6_rev1
  - New features:
    - Added support for NIST P-521 elliptic curve with CASPER driver.
    - Added support for using multiple elliptic curves at once with CASPER driver.
- 2.16.6
  - New features:
    - Ported mbedTLS 2.16.6 to SDK.
- 2.16.2_rev2
  - Bug fixes:
    - Add support for HASHCRYPT context switch check, Hashcrypt without context switch is not able to calculate SHA in parallel with AES. HW acceleration of SHA is disabled by default in MbedTLS integration, enabled on chip with context switch.
- 2.16.2_rev1
  - Bug fixes:
    - Add support for CTR_DRBG using AES-128 for crypto engines without AES-256 capability.
- 2.16.2
  - New features:
    - Ported mbedTLS 2.16.2 to SDK.
- 2.13.1_rev5
  - Bug fixes:
    - ecp_alt_ksdk.c fix CASPER port for ECJPAKE shortcut when points equal 1. This case is point addition and this shortcut follows original mbedtls_ecp_muladd() implementation which is required for ecjpake_ecp_add3().
- 2.13.1_rev4

  - New features:
    - Added support for NIST P-384 elliptic curve with CASPER driver.
- 2.13.1_rev3
  - Bug fixes:
    - Force align AES_CCM and AES_GCM self-test keys to fix unaligned key issue when using HW acceleration.
- 2.13.1_rev2
  - Bug fixes:
    - Disable default HW acceleration of SHA in parallel with AES.
- 2.13.1_rev1
  - Bug fixes:
    - Fixed incorrect macro check when skipping AES-192 or AES-256
- 2.13.1
  - New features:
    - Ported mbedTLS 2.13.1 to KSDK.
- 2.12.0_rev1
  - New features:
    - Added support for NIST P-256 elliptic curve with CASPER driver.
- 2.12.0
  - New features:
    - Ported mbedTLS 2.12.0 to KSDK.
- 2.9.0_rev2
  - New features:
    - Added support for Hashcrypt driver.
- 2.9.0_rev1
  - New features:
    - Added support for CASPER driver.
- 2.9.0
  - New features:
    - Ported mbedTLS 2.9.0 to KSDK.
- 2.6.0_rev2
  - Bug fixes:
    - ssl_cookie.c now uses SHA256 for COOKIE_MD (instead of original SHA224). Some hw crypto acceleration (such as CAU3) don't support SHA224 but all support SHA256.
- 2.6.0_rev1
  - Bug fixes:
    - ksdk_mbedtls.c bignum functions now read sign of input mbedtls_mpi at beginning of functions to properly support in place computations (when output bignum is the same as one of input bignums). Affected functions: mbedtls_mpi_mul_mpi(), mbedtls_mpi_mod-_mpi(), ecp_mul_comb().
- 2.6.0
  - New features:
    - Ported mbedTLS 2.6.0 to KSDK.
    - Added MBEDTLS_FREESCALE_FREERTOS_CALLOC_ALT to allow alternate implementation of pvPortCalloc() when using /middleware/mbedtls/port/ksdk/ksdk_-

mbedtls.c.

- 2.5.1_rev1
  - New features:
    - Added support for DCP driver.
- 2.5.1
  - New features:
    - Ported mbedTLS 2.5.1 to KSDK.
- 2.4.2_rev2
  - New features:
    - Added Curve25519 support for CAU3.
    - Added MBEDTLS_ECP_MUL_MXZ_ALT configuration parameter enabling overloading of ecp_mul_mxz().
- 2.4.2_rev1
  - New features:
    - Added support for CAU3 driver.
    - Added new files:
    - /middleware/mbedtls/port/ksdk/des_alt.c - contains regular software implementation of DES algorithm with added MBEDTLS_DES3_SETKEY_DEC_ALT and MBEDTLS_DES3_SETKEY_ENC_ALT config parameters.
    - /middleware/mbedtls/port/ksdk/des_alt.h - contains modified mbedtls_des_context and mbedtls_des3_context structures.
    - Added MBEDTLS_DES3_SETKEY_DEC_ALT configuration parameter enabling reloading of mbedtls_des3_set2key_dec() and mbedtls_des3_set3key_dec().
    - Added MBEDTLS_DES3_SETKEY_ENC_ALT configuration parameter enabling reloading of mbedtls_des3_set2key_enc() and mbedtls_des3_set3key_enc().
- 2.4.2
  - New features:
    - Ported mbedTLS 2.4.2 to KSDK 2.0.0.
    - Added CRYPTO_InitHardware() function.
    - Added new file:
      - /middleware/mbedtls/port/ksdk/ksdk_mbedtls.h - contains declaration of CRYPTO_InitHardware() function and should be included in applications.
- 2.3.0_rev1
  - New features:
    - Added support for CAAM driver.
    - In LTC-specific wrapper, allocate temporary integers from heap in one large block.
- 2.3.0
  - New features:
    - Ported mbedTLS 2.3.0 to KSDK 2.0.0.
- 2.2.1
  - New features:
    - Ported mbedTLS 2.2.1 to KSDK 2.0.0.
    - Added support of MMCAU cryptographic acceleration module. Accelerated MD5, SHA, AES, and DES.
    - Added support of LTC cryptographic acceleration module. Accelerated AES, DES, and

PKHA.
* Added new files:
* /middleware/mbedtls/port/ksdk/ksdk_mbedtls.c - alternative implementation of cryptographic algorithm functions using LTC and MMCAU module drivers.
* /middleware/mbedtls/port/ksdk/ksdk_mbedtls_config.h - configuration settings used by mbedTLS KSDK bare metal examples.
* Added mbedTLS KSDK bare-metal examples:
  - /boards/<board name>/demo_apps/mbedtls/mbedtls_benchmark - KSDK mbedTLS benchmark application.
  - /boards/<board name>/demo_apps/mbedtls/mbedtls_selftest - KSDK mbedTLS self-test application.
* Added MBEDTLS_GCM_CRYPT_ALT configuration parameter enabling reloading of mbedtls_gcm_crypt_and_tag().
* Added MBEDTLS_ECP_MUL_COMB_ALT to enable alternate implementation of ecp-_mul_comb().
* Added MBEDTLS_ECP_ADD_ALT configuration parameter enabling reloading of ecp-_add().
* Added MBEDTLS_DES_SETKEY_DEC_ALT configuration parameter enabling reloading of mbedtls_des_setkey_dec(), mbedtls_des3_set2key_dec() and mbedtls_des3-_set3key_dec().
* Added MBEDTLS_DES_SETKEY_ENC_ALT configuration parameter enabling reloading of mbedtls_des_setkey_enc(), mbedtls_des3_set2key_enc() and mbedtls_des3-_set3key_enc().
* Added MBEDTLS_DES_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_des_crypt_cbc().
* Added MBEDTLS_DES3_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_des3_crypt_cbc().
* Added MBEDTLS_AES_CRYPT_CBC_ALT configuration parameter enabling reloading of mbedtls_aes_crypt_cbc().
* Added MBEDTLS_AES_CRYPT_CTR_ALT configuration parameter enabling reloading of mbedtls_aes_crypt_ctr().
* Added MBEDTLS_CCM_CRYPT_ALT configuration parameter enabling reloading of mbedtls_ccm_encrypt_and_tag() and mbedtls_ccm_auth_decrypt().
* Added MBEDTLS_MPI_ADD_ABS_ALT configuration parameter enabling reloading of mbedtls_mpi_add_abs().
* Added MBEDTLS_MPI_SUB_ABS_ALT configuration parameter enabling reloading of mbedtls_mpi_sub_abs().
* Added MBEDTLS_MPI_EXP_MOD_ALT configuration parameter enabling reloading of mbedtls_mpi_exp_mod().
* Added MBEDTLS_MPI_MUL_MPI_ALT configuration parameter enabling reloading of mbedtls_mpi_mul_mpi().
* Added MBEDTLS_MPI_MOD_MPI_ALT configuration parameter enabling reloading of mbedtls_mpi_mod_mpi().
* Added MBEDTLS_MPI_GCD_ALT configuration parameter enabling reloading of mbedtls_mpi_gcd().

    - Added MBEDTLS_MPI_INV_MOD_ALT configuration parameter enabling reloading of mbedtls_mpi_inv_mod().
    - Added MBEDTLS_MPI_IS_PRIME_ALT configuration parameter enabling reloading of mbedtls_mpi_is_prime().
    - Added encrypt/decrypt mode to mbedtls_des_context and mbedtls_des3_context structure.
    - Added carriage return '\r' for mbedtls_printf() in self test functions.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.10.0.

- 2.10.0
  - New features and demos:
    - Implement the USB Host ECM.
      - Add new USB host example: usb_host_cdc_ecm
    - Add one new USB host audio example: usb_host_audio_unified.
    - eUSB support on EHCI.
    - Add L1 LPM low power feature on EHCI for device and host.
  - Improvement:
    - Enable cache maintenance in the usb_host_msd_fatfs, usb_device_msc_disk and usb_device_msc_ramdisk examples on the RT1040-EVK, RT1050-EVKB, RT1060-EVKC and RT1170-EVKB platforms.
    - Improve Host VNIC to be more compatible with other USB devices.
    - Add USB_DEVICE_CONFIG_SOF_NOTIFICATION for device stack.
    - Clear the pending FR_Swap during initialization to prevent the FR_Swap from affecting the later PD negotiation.
  - Bug fixes:
    - Fix on IP3511 driver that SETUP bit is cleared by mistake.
    - Fix on IP3516 driver that cannot do multiple ISO endpoints transfers at the same time.
    - Fix on IP3516 driver that the Token Done interrupt is cleared but the last completed transfer is not processed.
    - Fix on IP3516 driver that the transfer will continue when receiving a short packet.
    - Fix on host audio class driver that entities cannot be distinguished between recorder and speaker.
- 2.9.1
  - Improvement:
    - Update EHCI controller driver for basic support of eUSB.
    - Replace the hard code in audio cases with macro.
    - Uniform the Chapter9 for device lite cases.
- 2.9.0
  - Improvement:
    - Change ROOT2 as enabled by default in device stack.
    - Implement independent frequency adjustment for speaker and recorder of composite audio

unified demos.
    - Fix vulnerability for host stack. CVE number: CVE-2023-38749
    - Delete deprecated enet driver function for enet adapter.
- 2.8.4
  - Improvement:
    - Add the new netc adatper for the new netc driver.
    - Fix issues for USB device dfu and usb device msc when enable the macro USB_DEVICE_CONFIG_RETURN_VALUE_CHECK.
    - Change the header file including order for usb.h header.
    - Update the USB host audio class driver to fix the wrong output log.
    - Add the workaround on dev_hid_mouse_bm case for the errata TN00071.
    - Enable ROOT2 macro in USB device stack.
    - Use an unified definiton for the base address of RTxxxx platforms.
- 2.8.3
  - Improvement:
    - Update the EHCI controller driver to support the address convert for TCM.
    - Update the USB host EHCI controller driver to make sure the mutual exclusion access under multiple tasks' environment.
- 2.8.2
  - Improvement:
    - Fix noise issue of UAC 3.1, UAC 5.1, UAC 7.1 on usb audio speaker demo.
    - Fix the issue that incorrect PC behavior when ejecting USB MSC devices.
    - Update the EHCI controller driver to support RW610 that does not reply on PHY driver, especially for low power feature.
    - Update the USB_HostHelperParseAlternateSetting to fix the wrong interface parse.
    - Update dev_composite_hid_audio_unified_bm demo to suppport independent mute/unmute and volume control.
- 2.8.1
  - Improvement:
    - update USB audio demos to use audio component (components/audio).
    - Add the checking of function call return value.
    - Add audio multiple channels demo (usb_device_composite_audio_multi_ch_unified) on RT600 audio board.
    - Fix audio noise on sync mode and improve overflow/underflow checking method.
    - Support UAC 3.1, 5.1 and 7.1 on audio speaker demo.
    - Set USB device CDC demo not to depend on DTR setting from host.
    - Support MCUX toolchain on some RTxxxx platforms.
- 2.8.0
  - Improvement:
    - Fix the USB device stack vulnerability issues.
    - Update the audio PLL and FRO adjustment codes for audio examples in RTxxx, LPC54xxx and LPC55xxx.
    - Improve the USB PD AMS collision avoidance.
    - Improve IP3511 controller driver's dedicated ram allocation.
    - Change the USB_DATA_ALIGN_SIZE to 4 because the controller driver uses the

dedicated RAM to do memcpy.
  - New features:
    * Enable USB host audio recorder demo for mutilple boards.
- 2.7.0
  - Improvement:
    * Use new feeback solution and low latency playback for usb device speaker demo and unified demos. Add underflow and overflow protection.
    * Optimize hard code for usb audio demos.
    * Update Unconstrained Power field in the Sink Capabilities Message according to the external power state.
    * Fix CVE-2021-38258 and CVE-2021-38260
  - New features:
    * Enable USB host video demo for mutilple boards.
    * Enable USB device MTP demo for mutilple boards.
    * Add PPS message to usb pd stack.
- 2.6.1
  - Improvement:
    * rename sdcard as disk for all of sdcard demos. For ramdisk demos, they are not changed.
    * add wrapper for all of disk demos to support emmc.
- 2.6.0
  - Improvement:
    * Added more ufi event to support dynamic sdcard capacity.
    * Passed MISRA-2012 mandatory and required rules.
      · Except rule 17.2 in host hub and otg stack.
      · Except rule 5.1, rule 5.4, rule 21.1 and rule 21.2.
    * Re-implemented USB components and supported NPW.
    * Improved IP3511 controller driver's cancelling transfer function.
    * Enabled the audio2.0 defaultly for device audio demos.
    * Enabled the host audio2.0 function in host audio class driver and host audio speaker demo.
  - New features:
    * enable two USB controllers in one USB host mouse demo which named as host_hid_-mouse_dual.
    * enable UAC 5.1 for usb device audio speaker demo.
- 2.5.0
  - Improvement:
    * Integrated sdk components (OSA, Timer, GPIO and serial_manager) to USB stack and demos.
    * Improved the ip3511 driver throughput.
    * Improved audio initialization codes after SDK audio drivers update.
    * Improved auido to support the audio2.0 in win10.
    * Add one "enumeration fail" callback event to host stack.
- 2.4.2
  - Improvement:
    * Put the USB controller data and transfer buffer to noncache section, removed the setting that sets the whole ocram and sdram as noncached.

    - Separated composite audio examples' channel,sample rate,format parameters from commom macro to in dedicated macro and out dedicated macro.
    - replaced USB_PrepareData with USB_AudioRecorderGetBuffer.
- 2.4.1
  - New features:
    - Added enumeration fail callback to host stack when the attached device's enumeration failed.
- 2.4.0
  - Improvement:
    - Device Charger Detection (DCD) software architecture was refactored.
  - New features:
    - Enabled Device Charger Detection (DCD) on RT1060.
    - Enabled Device Charger Detection on RT600.
    - Enabled host battery charger function on RT600.
- 2.3.0
  - New features:
    - Added host video camera support. example: usb_host_video_camera
    - Added a new device example. example: usb_device_composite_cdc_hid_audio_unified
- 2.2.0
  - New features:
    - Added device DFU support.
    - Supported OM13790DOCK on LPCXpresso54018.
    - Added multiple logical unit support in msc class driver, updated usb_device_lba_information_struct_t to support this.
    - Supported multiple transfers for host ISO on IP3516HS.
  - Bug fixes:
    - Fixed device ip3511 prime data length than maxpacket size issue.
    - Initialized interval attribute in usb_device_endpoint_struct_t/usb_device_endpoint_init_struct_t.
    - Removed unnecessary header file in device CDC class driver, removed unnecessary usb_echo, and added DEBUG macro for necessary usb_echo in device CDC class driver.
    - Fixed device IP3511HS unfinished interrupt transfer missing issue.
- 2.1.0
  - New features:
    - Added host RNDIS support. example: lwip_dhcp_usb
    - Enabled USB 3.0 support on device stack.
    - Power Delivery feature: Added OM13790HOST support; Added auto policy feature; Printed e-marked cable information;
- 2.0.1
  - Bug fixes:
    - Fixed some USB issues: Fixed MSC CV test failed in MSC examples.
    - Changed audio codec interfaces.
- 2.0.0
  - New features:
    - PTN5110N support.

  - Bug fix:
    - Added some comments, fixed some minor USB issues.
- 1.9.0
  - New features:
    - Examples:
      - usb_pd_alt_mode_dp_host
- 1.8.2
  - Updated license.
- 1.8.1
  - Bug fix:
    - Verified some hardware issues, support aruba_flashless.
- 1.8.0
  - New features:
    - Examples:
      - usb_device_composite_cdc_vcom_cdc_vcom
      - usb_device_composite_hid_audio_unified
      - usb_pd_sink_battery
      - Changed usb_pd_battery to usb_pd_charger_battery.
  - Bug fix:
    - Code clean up, removed some irrelevant code.
- 1.7.0
  - New features:
    - USB PD stack support.
  - Examples:
    - usb_pd
    - usb_pd_battery
    - usb_pd_source_charger
- 1.6.3
  - Bug fix: -IP3511_HS driver control transfer sequence issue, enabled 3511 ip cv test.
- 1.6.2
  - New features:
    - Multi instance support.
- 1.6.1
  - New features:
  - Changed the struct variable address method for device_video_virtual_camera and host_phdc-_manager.
- 1.6.0
  - New features:
    - Supported Device Charger Detect feature on usb_device_hid_mouse.
- 1.5.0
  - New features:
    - Supported controllers
      - OHCI (Full Speed, Host mode)
      - IP3516 (High Speed, Host mode)
      - IP3511 (High Speed, Device mode)

        - Examples:
            - usb_lpm_device_hid_mouse
            - usb_lpm_device_hid_mouse_lite
            - usb_lpm_host_hid_mouse
- 1.4.0
  - New features:
    - Examples:
      - usb_device_hid_mouse/freertos_static
      - usb_suspend_resume_device_hid_mouse_lite
- 1.3.0
  - New features:
    - Supported roles
      - OTG
    - Supported classes
      - CDC RNDIS
    - Examples
      - usb_otg_hid_mouse
      - usb_device_cdc_vnic
      - usb_suspend_resume_device_hid_mouse
      - usb_suspend_resume_host_hid_mouse
- 1.2.0
  - New features:
    - Supported controllers
      - LPC IP3511 (Full Speed, Device mode)
- 1.1.0
  - Bug fix:
    - Fixed some issues in USB certification.
    - Changed VID and Manufacturer string to NXP.
  - New features:
    - Supported classes
      - Pinter
    - Examples:
      - usb_device_composite_cdc_msc_sdcard
      - usb_device_printer_virtual_plain_text
      - usb_host_printer_plain_text
- 1.0.1
  - Bug fix:
    - Improved the efficiency of device audio speaker by changing the transfer mode from interrupt to DMA, thus providing the ability to eliminate the periodic noise.
- 1.0.0
  - New features:
    - Supported roles
      - Device
      - Host
    - Supported controllers:

  - KHCI (Full Speed)
  - EHCI (High Speed)
* Supported classes:
  - AUDIO
  - CCID
  - CDC
  - HID
  - MSC
  - PHDC
  - VIDEO
* Examples:
  - usb_device_audio_generator
  - usb_device_audio_speaker
  - usb_device_ccid_smart_card
  - usb_device_cdc_vcom
  - usb_device_cdc_vnic
  - usb_device_composite_cdc_msc
  - usb_device_composite_hid_audio
  - usb_device_composite_hid_mouse_hid_keyboard
  - usb_device_hid_generic
  - usb_device_hid_mouse
  - usb_device_msc_ramdisk
  - usb_device_msc_sdcard
  - usb_device_phdc_weighscale
  - usb_device_video_flexio_ov7670
  - usb_device_video_virtual_camera
  - usb_host_audio_speaker
  - usb_host_cdc
  - usb_host_hid_generic
  - usb_host_hid_mouse
  - usb_host_hid_mouse_keyboard
  - usb_host_msd_command
  - usb_host_msd_fatfs
  - usb_host_phdc_manager
  - usb_keyboard2mouse
  - usb_pin_detect_hid_mouse

3 Component Change Log

CODEC

The current codec common driver version is 2.3.1.

- 2.3.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 16.1,16.3.
- 2.3.0
  - Improvements
    - Added enum _codec_volume_capability for CODEC_SetVolume/CODEC_SetMute to cover more volume configurations.
- 2.2.2
  - Bug Fixes
    - Fixed the typo in codec common driver.
- 2.2.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.2.0
  - Improvements
    - Used HAL_CODEC_HANDLER_SIZE which is determined by low level driver instead of use CODEC_HANDLE_SIZE for the codec device handle definition.
- 2.1.1
  - Improvements
    - Supported all of the codec in the codec adapter.
    - Modified the codec handle definition to improve user experience.
    - Modified the capability member type from entity to pointer in codec handle.
  - Bug Fixes
    - Fixed the Coverity issue regrading array compared agaist 0.
- 2.1.0
  - Deprecated APIs
    - CODEC_GetMappedFormatBits
    - CODEC_I2C_WriteReg
    - CODEC_I2C_ReadReg
    - CODEC_I2C_ModifyReg
    - CODEC_SetEncoding
  - new APIs
    - CODEC_SetPower
    - CODEC_SetVolume
    - CODEC_SetMute
    - CODEC_SetPlay
    - CODEC_SetRecord
    - CODEC_SetRecordChannel

        * CODEC_ModuleControl
    - new features
        * Removed duplicate members in codec_handle_t and codec_config_t.
        * Added codec_config_t pointer in codec_handle_t.
        * Added codec capability flag in codec_handle_t.
        * Used codec adapter instead of function opinter in codec common driver.
- 2.0.1
    - Added delayMs function pointer in codec handle.
- 2.0.0
    - Initial version.

WM8904

The current wm8904 driver version is 2.5.1.

- 2.5.1
    - Bug Fixes
        * Fixed invalid clock divider issue generated form WM8904_SetMasterClock api
        * Replace '__REV16' with general implementation to swap bytes in a short variable.
- 2.5.0
    - Improvements
        * Added master clock configuration support in function WM8904_SetAudioFormat.
        * Align the sysclk paramter definition for the WM8904_SetAudioFormat/WM8904_SetMasterClock.
        * Added api WM8904_SetDACVolume to support adjust DAC volume.
        * Fixed the MISRA-2012 violation of 12.2, 10.3.
- 2.4.4
    - Bug Fixes
        * Added the 11.025kHz/22.05kHz/44.1kHz samplerate support on codec WM8904.
        * Fixed the MISRA-2012 violation of 4.7.
- 2.4.3
    - Bug Fixes
        * Fixed the MISRA-2012 violations.
            · Fixed rule 8.6, 9.3, 10.1, 10.3, 10.4, 10.7, 10.8, 11.8, 11.9, 14.4, 16.1, 16.3, 16.4, 17.7, 20.9.
- 2.4.2
    - Bug Fixes
        * Corrected the volume setting function behavior in wm8904 driver, support range align with its specification range.
        * Corrected the volume setting function behavior in wm8904 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.4.1
    - Bug Fixes
        * Fixed the bit width reigster field overwritten issue.

- 2.4.0
  - New features
    - Added fll support in wm8904 driver.
- 2.3.0
  - Improvements
    - Added new API WM8904_SetMasterClock to support BCLK/LRCLK output mode.
- 2.1.0
  - new APIs
    - WM8904_ReadRegister
    - WM8904_WriteRegister
    - WM8904_ModifyRegister
    - WM8904_SetRecord
    - WM8904_SetPlay
    - WM8904_SetRecordChannel
    - WM8904_SetModulePower
    - WM8904_SetChannelVolume
    - WM8904_SetChannelMute
  - New features
    - Removed dependency on codec common driver.
    - Added dependency on codec i2c.
  - Bug Fixes
    - Fixed unchecked return value in WM8904_Deinit.
    - Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.3
  - Bug Fixes
    - Fixed issue that wm8904 register access function truncated return value.
- 2.0.2
  - Bug Fixes
    - Fixed using uninitialized value format.fsRatio when calling WM8904_UpdateFormate.
- 2.0.1
  - Added WM8904_CheckAudioFormat API.
  - Changed the second parameter's name of WM8904_SetAudioFormat to sysclk.
- 2.0.0
  - Initial version.

.1 WM8960

The current wm8960 driver version is 2.2.4.

- 2.2.4
  - Improvements
    - Remove CODEC_I2C_Deinit in WM8960_Deinit.
- 2.2.3
  - Improvements

        * Reinitialise I2C in Deinit function.
- 2.2.2
  - Bug fixes
    * Fixed violations of MISRA C-2012 rule 10.3.
- 2.2.1
  - Bug fixes
    * Improved the internal PLL fatctor calculation formula.
- 2.2.0
  - Improvements
    * Added masterClock member in wm8960_config_t to support wm8960 master mode.
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 4.7, 5.8, 10.3, 10.4, 12.2, 14.4.
    * Added the bit clock divider configuration when wm8960 act as master.
- 2.1.3
  - Bug Fixes
    * Fixed the issue that WM8960 had no ack when performing write register by updating the byte count to be written.
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.1.2
  - Improvements
    * Enabled the class D output in WM8960_Init.
  - Bug Fixes
    * Corrected the volume setting function behavior in wm8960 driver, support range aligned with its specification range.
    * Corrected the volume setting function behavior in wm8960 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.1.1
  - Improvements
    * Removed useless bit clock divider configuration in function WM8960_ConfigDataFormat.
- 2.1.0
  - Improvements
    * Added new API WM8960_SetPlay.
    * Fixed error status overwrite issue in WM8960_ConfigDataFormat function.
    * Removed dependency on codec common driver.
    * Added dependency on codec i2c.
  - Bug Fixes
    * Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.2
  - Removed bit width hard code setting in function WM8960_SetProtocol.
- 2.0.1
  - Corrected the bclk divider calculation.
- 2.0.0
  - Initial version.

SGTL5000

The current sgtl5000 driver version is 2.1.1.

- 2.1.1
  - Improvements
    - Corrected the volume setting function behavior in SGTL5000 driver, support range align with its specification range.
    - Corrected the volume setting function behavior in SGTL5000 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.1.0
  - Improvements
    - Added API SGTL_SetPlay/SGTL_SetRecord.
    - Removed dependency on codec common driver.
    - Added dependency on codec i2c.
    - Fixed divison or modulo by zero issue in SGTL_ConfigDataFormat function.
  - Bug Fixes
    - Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.0
  - Initial version.

DA7212

The current da7212 driver version is 2.3.1.

- 2.3.1
  - Improvements
    - Disable ramp setting in init function.
- 2.3.0
  - Improvements
    - Add input sourece select in init function.
- 2.2.3
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.4, 4.7.
- 2.2.2
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.6, 9.3, 10.1, 10.3, 10.4, 10.7, 10.9, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.2.1
  - Improvements
    - Corrected the volume setting function behavior in DA7212 driver, support range align with its specification range.

        * Corrected the volume setting function behavior in DA7212 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.2.0
  - Improvements
    * Added bclk invert parameter in the format structure.
    * Added API DA7212_SetMasterModeBits/DA7212_SetPLLConfig.
    * Added pll/sysClkSource parameters in the da7212 configuration structure.
    * Disbaled PLL by default.
- 2.1.0
  - Improvements
    * Removed dependency on codec common driver.
    * Added dependency on codec i2c.
  - Bug Fixes
    * Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.0
  - Initial version.

CS42888

The current cs42888 driver version is 2.1.3

- 2.1.3
  - Improvements
    * Removed the assertion for codec reset function pointer.
- 2.1.2
  - Improvements
    * Corrected the volume setting function behavior in CS42888 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 4.7, 10.3, 8.3, 10.7, 17.7.
    * Corrected the channel index during setting AIN volume in CS42888_Init.
- 2.1.1
  - Improvements
    * Used software delay with delayMs pointer not provided by application.
    * Fixed error status overwrite issue in CS42888_Init function.
    * Removed dependency on codec common driver.
    * Added API CS42888_SelectFunctionalMode/CS42888_SetChannelMute.
    * Added dependency on codec i2c.
- 2.1.0
  - Improvements
    * Unified CS42888 codec driver interface.
  - Bug Fixes
    * Corrected the ADC/DAC functional mode macro definitaion.
    * Added TDM and OLM mode support in the function CS42888_SetProtocol.

- 2.0.0
  - Initial version.

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

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support