MCUXpresso SDK Release Notes Supporting evkmimx8mp

Change Logs

NXP Semiconductors

Contents

Driver Change Log

Title

Page No.

Middleware Change Log

Component Change Log

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.2.0.

- 2.2.0
  - New feature
    - Add new API CLOCK_GetEnetAxiFreq() to get frequency for ENET.
- 2.1.1
  - Improvements
    - Make driver compilable with GNU C++ Compiler.
- 2.1.0
  - New feature
    - Add new API CLOCK_GetClockRootFreq() to get frequency for peripherals.
- 2.0.2
  - Improvements
    - Make driver aarch64 compatible.
    - Add Peripheral Clock definition.
- 2.0.1
  - Bug Fixes
    - Corrected and added clock information for IOMUX and IPMUX.
- 2.0.0
  - initial version.

AUDIOMIX

The current AUDIOMIX driver version is 2.0.2.

- 2.0.2
  - Improvements
    - Make AUDIOMIX driver to be aarch64 compatible.
- 2.0.1
  - Bug Fixes
    - Corrected the logic to wait the audio pll locked.
- 2.0.0
  - initial version.

IOMUXC

The current IOMUXC driver version is 2.0.5.

- 2.0.5

  - Improvements
    - Make iomuxc driver to be aarch64 compatible.
- 2.0.4
  - Improvements
    - Updated some pins function name.
    - Removed SRC_EARLY_RESET and SRC_SYSTEM_RESET pin function.
- 2.0.3
  - Improvements
    - Updated some pins function name.
    - Removed SRC_BOOT_MODE4/5 pin function.
- 2.0.2
  - Bug Fixes
    - Removed the SAI5 functions from SAI1 IO pad.
- 2.0.1
  - Doxygen improvement.
- 2.0.0
  - Initial version.

MEMORY

The current MEMORY driver version is 2.0.1.

- 2.0.1
  - Improvements
    - Make driver aarch64 compatible.
- 2.0.0
  - Initial version.

ECSPI_CMSIS

Current ecspi_cmsis driver version is 2.3

- 2.3
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 10.3, 11.1, 11.8, 16.1, 16.3.
- 2.2
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 11.9, 14.4, 16.1, 16.3, 16.4, 20.7, 20.9.
- 2.1
  - 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.

- 2.0
  - Initial version.

ENET

Current ENET CMSIS driver version is 2.3

- 2.3
  - Bug Fixes
    - Updated the driver capabilities aligned with the actual situation.
- 2.2
  - New Features
    - Added code to deal with 1G enet and RGMII interface configuration in cmsis enet driver.
- 2.1
  - Bug Fixes
    - Fixed the wrong logic to control cache macro.
- 2.0
  - Initial version.

I2C

Current I2C CMSIS driver version is 2.2

- 2.2
  - Bug Fixes
    - Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 11.9, 14.4, 16.1, 16.3, 16.4, 20.7, 20.9.
- 2.1
  - Bug Fixes
    - Fixed the bug that in PowerControl, module should be reset first by calling I2C_MasterInit and I2C_SlaveInit.
- 2.0
  - Initial version.

UART

The current UART CMSIS driver version is 2.1

- 2.1
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules: 10.1,10.3,10.4,11.1,11.9,14.4,15.7,16.1,16.-6,20.9.
- 2.0

  - 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.

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.

eASRC

The current eASRC driver version is 2.0.6.

- 2.0.6
  - Improvements
    * Updated use of definitions in the function ASRC_SetSlotConfig and ASRC_EnableContextSlot.
- 2.0.5
  - Improvements
    * Fixed the typo and multichannel configuration issues in asrc driver.
    * Updated the asrc firmware table for better THD performance.
- 2.0.4
  - Improvements
    * Corrected the data width support by IEC60958,
    * Updated the default stage1 result to float to avoid convert result distortion.

- 2.0.3
  - Bug Fixes
    - Corrected the btis shift index used in function ASRC_TransferBlocking.
    - Ensured the resampler and prefilter are not in bypass mode when the convertion require the function.
- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 14.3, 14.4, 16.3, 16.1, 16.8, 17.7.
- 2.0.1
  - Bug Fixes
    - Fixed the context id hard code issue in the function ASRC_TransferInCreateHandleSDMA/ASRC_TransferOutCreateHandleSDMA.
  - Improvements
    - Added support for the data size bigger than 64K in sdma driver.
- 2.0.0
  - Initial version.

ECSPI

The current eCSPI driver version is 2.3.3.

- 2.3.3
  - Bug Fixes
    - Fixed the txData from void * to const void * in transmit API
- 2.3.2
  - Improvements
    - Changed ECSPI_DUMMYDATA to 0x00.
- 2.3.1
  - Bug Fixes
    - Fixed ECSPI_GetInstance potential issue that return wrong instance number.
- 2.3.0
  - Bug Fixes
    - Fixed burst length issue,the burst length range shall range from 1-4096 bits, so the width shall be uint8_t rather than uint16_t.
- 2.2.0
  - Bug Fixes
    - Removed the useless channel configuration of waveform, since the waveform can not be configured when not using the exchange bit(ECSPIx_CONREG[XCH]) for the transfer.
    - Fixed violations of MISRA C-2012 rules: 10.1, 11.9, 8.4.
- 2.1.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 10.1, 10.3, 10.4, 11.9, 14.4, 15.7, 17.7.
- 2.1.0

  - Improvements
    - Added timeout mechanism when waiting certain states in transfer driver.
- 2.0.2
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 10.1, 10.3, 10.4
- 2.0.1
  - Bug Fixes
    - Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
    - Fixed sign-compare warning in ECSPI_SendTransfer.
- 2.0.0
  - Initial version.

ENET

The current ENET driver version is 2.9.1.

- 2.9.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 8.4, 10.4.
- 2.9.0
  - Bug Fixes
    - Enabled collection of transfer statistics, so the function ENET_GetStatistics does not always return zeroes.
  - New Features
    - Added new function ENET_EnableStatistics to enable/disable collection of transfer statistics.
    - Added new function ENET_ResetStatistics to reset transfer statistics.
  - Improvements
    - Renamed the function ENET_ResetHareware to ENET_ResetHardware.
- 2.8.0
  - New Features
    - Added the function to reset hardware on certain devices.
- 2.7.1
  - Bug Fixes
    - Fixed the issue that free wrong buffer address when one frame stores in multiple buffers and memory pool is not enough to allocate these buffers to receive one complete frame.
- 2.7.0
  - Improvements
    - Deleted deprecated zero copy Tx/Rx functions and set callback function which can be configured in ENET_Init.
    - Moved the Rx zero copy buffer allocation to Rx BD initialization function to reduce unnecessary looping code.
  - Bug Fixes

    - Fixed the issue that predefined Rx buffers which should not be used when enabling Rx zero copy are still be handled by cache operation, it causes hardfault on some platforms.
    - Fixed the issue that zero-copy Rx function doesn't check Rx length of 0 in the BD with EMPTY bit is 0, it may occur in the corner case reported by customer. Not sure how it turns out, consider it as an ENET IP issue and drop this abnormal BD.
- 2.6.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 11.6.
- 2.6.2
  - Improvements
    - Changed ENET1_MAC0_Rx_Tx_Done0_DriverIRQHandler/ENET1_MAC0_Rx_Tx_Done1_DriverIRQHandler to ENET1_MAC0_Rx_Tx_Done1_DriverIRQHandler/ENET1_MAC0_Rx_Tx_Done2_DriverIRQHandler which represent ring 1 and ring 2.
- 2.6.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 10.7, 11.6, 11.8.
- 2.6.0
  - Improvements
    - Added MDIO access wrapper APIs for ease of use.
    - Fixed the build warning introduced by 64-bit compatibility patch.
- 2.5.4
  - Improvements
    - Made the driver compatible with 64-bit platforms.
- 2.5.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 11.6.
- 2.5.2
  - Improvements
    - Updated the TXIC/RXIC register handling code according to the new header file.
- 2.5.1
  - Bug Fixes
    - Fixed document typo.
- 2.5.0
  - Bug Fixes
    - Fixed the SendFrame/SendFrameZeroCopy functions issue with scattered buffers.
    - Updated the formula of MDC calculation.
    - Used a feature macro to distinguish the old IP design from the new design, because old IP design always reads a value zero from ATCR->CAPTURE bit. For old IP, driver caculates and wait the necessary delay cycles after setting ATCR->CAPTURE then gets the timestamp value.
  - New Features
    - Added new zero copy Tx/Rx function.
    - New zero copy Tx function combines scattered and contiguous Tx buffer in one API, it also supports more Tx featrues which buffer descriptor supports but previous Tx function doesn't support.

    - New zero copy Rx function use dynamic buffer mechanism and simpler interface.
  - Improvements
    - Corrected the interrupt handler for PTP timestamp IRQ and PTP1588 event IRQ since platform difference.
    - Added missing IRQ handlers for PTP1588 events on some platforms.
    - Corrected the max Tx frame length verification, it will not depend on a fixed macro. The ENET_FRAME_MAX_FRAMELEN is only an default value for driver, application can configure it. Driver caculates the limitation with the max frame length in register which may takes extended 4 or 8 bytes VLAN tag if VLAN/SVLAN enables.
    - Deleted deprecated Clause 45 read/write legacy APIs.
- 2.4.3
  - Improvements
    - Aligned the IRQ handler name with header file.
- 2.4.2
  - Bug Fixes
    - Fixed the MISRA issue of speculative out-of-bounds access.
- 2.4.1
  - Bug Fixes
    - Fixed the PTP time capture issue.
- 2.4.0
  - Improvements
    - Exposed API ENET_ReclaimTxDescriptor for user application to relaim tx descriptors in their application.
    - Added counter to record multicast hash conflict in struct _enet_handle, improved the situation that one multicast group could be left by other conflict multicast address left operation.
    - Improved concurrent usage of relaim and send frame operation.
- 2.3.4
  - Bug Fixes
    - Fixed the issue that interrupt handler only checks the interrupt event flag but not checks interrupt mask flag.
- 2.3.3
  - Bug Fixes
    - Fixed the issue that some compilers may choose the memcpy with 4-bit aligned address limitation due to the type of address pointer is 'unsigned int *', the data address doesn't have to be 4-bit aligned.
- 2.3.2
  - New Features
    - Added the feature that ENET driver can be used in the platform which integrates both 10/100M and 1G ENET IP.
    - Deleted duplicated code about ARM errata 838869 in first/second level IRQ handler.
- 2.3.1
  - Improvements
    - Added function pointer checking in IRQ handler to make sure code can be used even it runs into the interrupt when the second level interupt handler is NULL.

- 2.3.0
  - Bug Fixes
    - Fixed the issue that clause 45 MDIO read/write API doesn't check the transmission over status between two transmissions.
    - Fixed violations of the MISRA C-2012 rules 2.2,10.3,10.4,10.7,11.6,11.8,13.5,14.4,15.-7,17.7.
  - New Features
    - Added APIs to support send/receive frame with Zero-Copy.
  - Improvements
    - Separated the clock configuration from module configuration when init and deinit.
    - Added functions to set second level interrupt handler.
    - Provided new function to get 1588 timer count without disabling interrupt.
    - Improved timestamp controlling, deleted all old timestamp management APIs and data structures.
    - Merged the single/multiple ring(s) APIs, now these APIs can handle both.
    - Used base and index to control buffer descriptor, aligned with qos and lpc enet driver.
- 2.2.6
  - Bug Fixes
    - Updated MII speed formula referring to the manual.
- 2.2.5
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.7, 11.6, 11.9, 13.5, 14.4, 16.4, 17.7, 21.15, 3.1, 8.4.
    - Changed to use ARRAY_SIZE(s_enetBases) as the array size for s_ENETHandle, fixed the hardfault issue for using some ENET instance when ARRAY_SIZE(s_enetBases) is not same as FSL_FEATURE_SOC_ENET_COUNT.
- 2.2.4
  - Improvements
    - Added call to Data Synchronization Barrier instruction before activating Tx/Rx buffer descriptor to ensure previous data update is completed.
    - Improved ENET_TransmitIRQHandler to store timestamps for multiple transmit buffer descriptors.
    - Bug Fixes
    - Fixed the issue that ENET_Ptp1588GetTimer did not handle the timer wrap situation.
- 2.2.3
  - Improvements
    - Improved data buffer cache maintenance in the ENET driver.
- 2.2.2
  - New Features
    - Added APIs for extended multi-ring support.
    - Added the AVB configure API for extended AVB feature support.
- 2.2.1
  - Improvements
    - Changed the input data pointer attribute to const in ENET_SendFrame().
- 2.1.1

  - New Features
    - Added the extended MDIO IEEE802.3 Clause 45 MDIO format SMI command APIs.
    - Added the extended interrupt coalescing feature.
  - Improvements
    - Combined all storage operations in the ENET_Init to ENET_SetHandler API.
- 2.0.1
  - Bug Fixes
    - Used direct transmit busy check when doing data transmit.
  - Miscellaneous Changes
    - Updated IRQ handler work flow.
    - Changed the TX/RX interrupt macro from kENET_RxByteInterrupt to kENET_RxBufferInterrupt, from kENET_TxByteInterrupt to kENET_TxBufferInterrupt.
    - Deleted unnecessary parameters in ENET handler.
- 2.0.0
  - Initial version.

FLEXCAN

The current FLEXCAN driver version is 2.11.6.

- 2.11.6
  - Bug Fixes
    - Fixed ERRATA_9595 FLEXCAN_EnterFreezeMode() may result to bus fault on some platform.
- 2.11.5
  - Bug Fixes
    - Fixed flexcan_memset() crash under high optimization compilation.
- 2.11.4
  - Improvements
    - Update CANFD max bitrate to 10Mbps on MCXNx3x and MCXNx4x.
    - Release peripheral from reset if necessary in init function.
- 2.11.3
  - Bug Fixes
    - Fixed FLEXCAN_TransferReceiveEnhancedFifoEDMA() compile error with DMA3.
- 2.11.2
  - Bug Fixes
    - Fixed bug that timestamp in flexcan_handle_t not updated when RX overflow happens.
- 2.11.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1.
- 2.11.0
  - Bug Fixes
    - Fixed wrong base address argument in FLEXCAN2 IRQ Handler.
  - Improvements

    - Add API to determine if the instance supports CAN FD mode at run time.
- 2.10.1
  - Bug Fixes
    - Fixed HIS CCM issue.
    - Fixed RTOS issue by adding protection to read-modify-write operations on interrupt enable/disable API.
- 2.10.0
  - Improvements
    - Update driver to make it able to support devices which has more than 64 8bytes MBs.
    - Update CAN FD transfer APIs to make them set/get edl bit according to frame content, which can make them compatible with classic CAN.
- 2.9.2
  - Bug Fixes
    - Fixed the issue that FLEXCAN_CheckUnhandleInterruptEvents() can't detecting the exist enhanced RX FIFO interrupt status.
    - Fixed the issue that FLEXCAN_ReadPNWakeUpMB() does not return fail even no existing valid wake-up frame.
    - Fixed the issue that FLEXCAN_ReadEnhancedRxFifo() may clear bits other than the data available bit.
    - Fixed violations of the MISRA C-2012 rules 10.4, 10.8.
  - Improvements
    - Return kStatus_FLEXCAN_RxFifoDisabled instead of kStatus_Fail when read FIFO fail during IRQ handler.
    - Remove unreachable code from timing calculates APIs.
    - Update Enhanced Rx FIFO handler to make it deal with underflow/overflow status first.
- 2.9.1
  - Bug Fixes
    - Fixed the issue that FLEXCAN_TransferReceiveEnhancedFifoBlocking() API clearing Fifo data available flag more than once.
    - Fixed the issue that entering FLEXCAN_SubHandlerForEhancedRxFifo() even if Enhanced Rx fifo interrupts are not enabled.
    - Fixed the issue that FLEXCAN_TransferReceiveEnhancedFifoEDMA() update handle even if previous Rx FIFO receive not finished.
    - Fixed the issue that FLEXCAN_SetEnhancedRxFifoConfig() not configure the ERFCR[NFE] bits to the correct value.
    - Fixed the issue that FLEXCAN_ReceiveFifoEDMACallback() can't differentiate between Rx fifo and enhanced rx fifo.
    - Fixed the issue that FLEXCAN_TransferHandleIRQ() can't report Legacy Rx FIFO warning status.
- 2.9.0
  - Improvements
    - Add public set bit rate API to make driver easier to use.
    - Update Legacy Rx FIFO transfer APIs to make it support received multiple frames during one API call.
    - Optimized FLEXCAN_SubHandlerForDataTransfered() API in interrupt handling to

reduce the probability of packet loss.

- 2.8.7
  - Improvements
    - Initialized the EDMA configuration structure in the FLEXCAN EDMA driver.
- 2.8.6
  - Bug Fixes
    - Fix Coverity overrun issues in fsl_flexcan_edma driver.
- 2.8.5
  - Improvements
    - Make driver aarch64 compatible.
- 2.8.4
  - Bug Fixes
    - Fixed FlexCan_Errata_6032 to disable all interrupts.
- 2.8.3
  - Bug Fixes
    - Fixed an issue with the FLEXCAN_EnableInterrupts and FLEXCAN_DisableInterrupts interrupt enable bits in the CTRL1 register.
- 2.8.2
  - Bug Fixes
    - Fixed errors in timing calculations and simplify the calculation process.
    - Fixed issue of CBT and FDCBT register may write failure.
- 2.8.1
  - Bug Fixes
    - Fixed the issue of CAN FD three sampling points.
    - Added macro to support the devices that no MCR[SUPV] bit.
    - Remove unnecessary clear WMB operations.
- 2.8.0
  - Improvements
    - Update config configuration.
      - Added enableSupervisorMode member to support enable/disable Supervisor mode.
    - Simplified the algorithm in CAN FD improved timing APIs.
- 2.7.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.7.
- 2.7.0
  - Improvements
    - Update config configuration.
      - Added enablePretendedeNetworking member to support enable/disable Pretended Networking feature.
      - Added enableTransceiverDelayMeasure member to support enable/disable Transceiver Delay MeasurementPretended feature.
      - Added bitRate/bitRateFD member to work as baudRate/baudRateFD member union.
    - Rename all "baud" in code or comments to "bit" to align with the CAN spec.
    - Added Pretended Networking mode related APIs.
      - FLEXCAN_SetPNConfig

        - FLEXCAN_GetPNMatchCount
        - FLEXCAN_ReadPNWakeUpMB
      - Added support for Enhanced Rx FIFO.
      - Removed independent memory error interrupt/status APIs and put all interrupt/status control operation into FLEXCAN_EnableInterrupts/FLEXCAN_DisableInterrupts and FLEXCAN_GetStatusFlags/FLEXCAN_ClearStatusFlags APIs.
      - Update improved timing APIs to make it calculate improved timing according to CiA doc recommended.
        - FLEXCAN_CalculateImprovedTimingValues.
        - FLEXCAN_FDCalculateImprovedTimingValues.
      - Update FLEXCAN_SetBitRate/FLEXCAN_SetFDBitRate to added the use of enhanced timing registers.
- 2.6.2
  - Improvements
    - Add CANFD frame data length enumeration.
- 2.6.1
  - Bug Fixes
    - Fixed the issue of not fully initializing memory in FLEXCAN_Reset() API.
- 2.6.0
  - Improvements
    - Enable CANFD ISO mode in FLEXCAN_FDInit API.
    - Enable the transceiver delay compensation feature when enable FD operation and set bitrate switch.
    - Implementation memory error control in FLEXCAN_Init API.
    - Improve FLEXCAN_FDCalculateImprovedTimingValues API to get same value for FPRESDIV and PRESDIV.
    - Added memory error configuration for user.
      - enableMemoryErrorControl
      - enableNonCorrectableErrorEnterFreeze
    - Added memory error related APIs.
      - FLEXCAN_GetMemoryErrorReportStatus
      - FLEXCAN_GetMemoryErrorStatusFlags
      - FLEXCAN_ClearMemoryErrorStatusFlags
      - FLEXCAN_EnableMemoryErrorInterrupts
      - FLEXCAN_DisableMemoryErrorInterrupts
  - Bug Fixes
    - Fixed the issue of sent duff CAN frame after call FLEXCAN_FDInit() API.
- 2.5.2
  - Bug Fixes
    - Fixed the code error issue and simplified the algorithm in improved timing APIs.
      - The bit field in CTRL1 register couldn't calculate higher ideal SP, we set it as the lowest one(75%)
      - FLEXCAN_CalculateImprovedTimingValues
      - FLEXCAN_FDCalculateImprovedTimingValues
    - Fixed MISRA-C 2012 Rule 17.7 and 14.4.

  - Improvements
    - Pass EsrStatus to callback function when kStatus_FLEXCAN_ErrorStatus is comming.
- 2.5.1
  - Bug Fixes
    - Fixed the non-divisible case in improved timing APIs.
      - FLEXCAN_CalculateImprovedTimingValues
      - FLEXCAN_FDCalculateImprovedTimingValues
- 2.5.0
  - Bug Fixes
    - MISRA C-2012 issue check.
      - Fixed rules, containing: rule-10.1, rule-10.3, rule-10.4, rule-10.7, rule-10.8, rule-11.8, rule-12.2, rule-13.4, rule-14.4, rule-15.5, rule-15.6, rule-15.7, rule-16.4, rule-17.3, rule-5.8, rule-8.3, rule-8.5.
    - Fixed the issue that API FLEXCAN_SetFDRxMbConfig lacks inactive message buff.
    - Fixed the issue of Pa082 warning.
    - Fixed the issue of dead lock in the function of interruption handler.
    - Fixed the issue of Legacy Rx Fifo EDMA transfer data fail in evkmimxrt1060 and evkmimxrt1064.
    - Fixed the issue of setting CANFD Bit Rate Switch.
    - Fixed the issue of operating unknown pointer risk.
      - when used the pointer "handle->mbFrameBuf[mbIdx]" to update the timestamp in a short-live TX frame, the frame pointer became as unknown, the action of operating it would result in program stack destroyed.
    - Added assert to check current CAN clock source affected by other clock gates in current device.
      - In some chips, CAN clock sources could be selected by CCM. But for some clock sources affected by other clock gates, if user insisted on using that clock source, they had to open these gates at the same time. However, they should take into consideration the power consumption issue at system level. In RT10xx chips, CAN clock source 2 was affected by the clock gate of lpuart1. ERRATA ID: (ERR050235 in CCM).
  - Improvements
    - Implementation for new FLEXCAN with ECC feature able to exit Freeze mode.
    - Optimized the function of interruption handler.
    - Added two APIs for FLEXCAN EDMA driver.
      - FLEXCAN_PrepareTransfConfiguration
      - FLEXCAN_StartTransferDatafromRxFIFO
    - Added new API for FLEXCAN driver.
      - FLEXCAN_GetTimeStamp
      - For TX non-blocking API, we wrote the frame into mailbox only, so no need to register TX frame address to the pointer, and the timestamp could be updated into the new global variable handle->timestamp[mbIdx], the FLEXCAN driver provided a new API for user to get it by handle and index number after TX DONE Success.
      - FLEXCAN_EnterFreezeMode

        - FLEXCAN_ExitFreezeMode
      - Added new configuration for user.
        - disableSelfReception
        - enableListenOnlyMode
      - Renamed the two clock source enum macros based on CLKSRC bit field value directly.
        - The CLKSRC bit value had no property about Oscillator or Peripheral type in lots of devices, it acted as two different clock input source only, but the legacy enum macros name contained such property, that misled user to select incorrect CAN clock source.
      - Created two new enum macros for the FLEXCAN driver.
        - kFLEXCAN_ClkSrc0
        - kFLEXCAN_ClkSrc1
      - Deprecated two legacy enum macros for the FLEXCAN driver.
        - kFLEXCAN_ClkSrcOsc
        - kFLEXCAN_ClkSrcPeri
      - Changed the process flow for Remote request frame response..
        - Created a new enum macro for the FLEXCAN driver.
        - kStatus_FLEXCAN_RxRemote
      - Changed the process flow for kFLEXCAN_StateRxRemote state in the interrupt handler.
        - Should the TX frame not register to the pointer of frame handle, interrupt handler would not be able to read the remote response frame from the mail box to ram, so user should read the frame by manual from mail box after a complete remote frame transfer.
- 2.4.0
  - Bug Fixes
    - MISRA C-2012 issue check.
      - Fixed rules, containing: rule-12.1, rule-17.7, rule-16.4, rule-11.9, rule-8.4, rule-14.4, rule-10.8, rule-10.4, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-8.3, rule-12.2 and rule-16.1.
    - Fixed the issue that CANFD transfer data fail when bus baudrate is 30Khz.
    - Fixed the issue that ERR009595 does not folllow the ERRATA document.
    - Fixed code error for ERR006032 work around solution.
    - Fixed the Coverity issue of BAD_SHIFT in FLEXCAN.
    - Fixed the Repo build warning issue for variable without initial.
  - Improvements
    - Fixed the run fail issue of FlexCAN RemoteRequest UT Case.
    - Implementation all TX and RX transfering Timestamp used in FlexCAN demos.
    - Fixed the issue of UT Test Fail for CANFD payload size changed from 64BperMB to 8PerMB.
    - Implementation for improved timing API by baud rate.
- 2.3.2
  - Improvements
    - Implementation for ERR005959.
    - Implementation for ERR005829.
    - Implementation for ERR006032.

- 2.3.1
  - Bug Fixes
    - Added correct handle when kStatus_FLEXCAN_TxSwitchToRx is comming.
- 2.3.0
  - Improvements
    - Added self-wakeup support for STOP mode in the interrupt handling.
- 2.2.3
  - Bug Fixes
    - Fixed the issue of CANFD data phase's bit rate not set as expected.
- 2.2.2
  - Improvements
    - Added a time stamp feature and enable it in the interrupt_transfer example.
- 2.2.1
  - Improvements
    - Separated CANFD initialization API.
    - In the interrupt handling, fix the issue that the user cannot use the normal CAN API when with an FD.
- 2.2.0
  - Improvements
    - Added FSL_FEATURE_FLEXCAN_HAS_SUPPORT_ENGINE_CLK_SEL_REMOVE feature to support SoCs without CAN Engine Clock selection in FlexCAN module.
    - Added FlexCAN Serial Clock Operation to support i.MX SoCs.
- 2.1.0
  - Bug Fixes
    - Corrected the spelling error in the function name FLEXCAN_XXX().
    - Moved Freeze Enable/Disable setting from FLEXCAN_Enter/ExitFreezeMode() to FLEXCAN_Init().
    - Corrected wrong helper macro values.
  - Improvements
    - Hid FLEXCAN_Reset() from user.
    - Used NDEBUG macro to wrap FLEXCAN_IsMbOccupied() function instead of DEBUG macro.
- 2.0.0
  - Initial version.

GPC

The current GPC driver version is 2.2.0.

- 2.2.0
  - Improvements
    - Optimized the exited APIs to support extended IRQs and slots.
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules.

- 2.1.0
  - Improvements
    - Unified the register or bit fields' name which contains specific cortex M core information.
  - Bug Fixes
    - Fixed Coverity Out-of-bounds issue.
- 2.0.1
  - Improvements
    - Added parameters to enable/disable the WFI and DSM mask.
- 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.

I2C

The current I2C driver version is 2.0.7.

- 2.0.7
  - Bug Fixes
    - Fixed MISRA issues.
      - Fixed rules 8.4, 8.5.
- 2.0.6
  - Bug Fixes
    - Fixed the bug that, in I2C_MasterStop after the stop command is issued, the IBB flag should be cleared rather than set.
    - Fixed the bug that to clear kI2C_ArbitrationLostFlag and kI2C_IntPendingFlag, their bits should be written '0' rather than '1'.
- 2.0.5
  - Bug Fixes
    - Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
    - Fixed MISRA issues.

        - Fixed rules 10.1, 10.3, 10.4, 11.9, 14.4, 15.7, 16.4, 17.7.
    - Improvements
        - Updated the I2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
- 2.0.4
    - Bug Fixes
        - Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) did not support the situation that master transfer with subaddress and transfer data size being zero, which means no data followed by the subaddress.
- 2.0.3
    - Improvements
        - Improved code readability, added new static API I2C_WaitForStatusReady for the status flag wait, and changed to call I2C_WaitForStatusReady instead of polling flags with reading register.
- 2.0.2
    - Improvements
        - Added I2C_WATI_TIMEOUT macro to allow users to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.1
    - Bug Fixes
        - Added a proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Only supports write only or write+read with no start flag; does not support read only with no start flag.
- 2.0.0
    - Initial version.

PWM

The current PWM driver version is 2.0.1.

- 2.0.1
    - Bug Fixes
        - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.0
    - Initial version.

UART

The current UART driver version is 2.3.2.

- 2.3.2
    - Improvements
        - Make driver aarch64 compatible
- 2.3.1
    - Improvements

        * Use separate data for TX and RX in uart_transfer_t.
    - Bug Fixes
        * Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling UART_TransferReceiveNonBlocking, the received data count returned by UART_TransferGetReceiveCount is wrong.
- 2.3.0
    - Bug Fixes
        * Fixed DMA transfer blocking issue by enabling tx idle interrupt after DMA transmission finishes.
- 2.2.1
    - Bug Fixes
        * Fixed MISRA 2012 rule 10.4 violation.
- 2.2.0
    - New Features
        * Modified uart_config_t, UART_Init and UART_GetDefaultConfig APIs so that the RTS and CTS used for hardware flow control can be enabled during module initialization.
        * Added API UART_SetRxRTSWatermark so that the water mark level of RTS deassertion can be configured.
- 2.1.1
    - Bug Fixes
        * Fixed MISRA 8.5 violation.
- 2.1.0
    - Improvements
        * Added timeout mechanism when waiting for certain states in transfer driver.
- 2.0.2
    - Improvements
        * Added check for transmission complete in UART_WriteBlocking, UART_TransferHandleIRQ and UART_SendSDMACallback to ensure all the data would be sent out to bus.
        * Modified UART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
    - Bug Fixes
        * Fixed MISRA issues.
            · Fixed rules 10.1, 10.3, 10.4, 10.6, 10.7, 10.8, 11.9, 14.4.
- 2.0.1
    - Bug Fixes
        * Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
- 2.0.0
    - Initial version.

UART_FREERTOS

The current UART_FREERTOS driver version is 2.1.1.

- 2.1.1
  - Refer UART driver change log 2.0.0 to 2.1.1.

UART_SDMA

The current UART_SDMA driver version is 2.3.0.

- 2.3.0
  - Refer UART driver change log 2.0.0 to 2.1.1, 2.3.0.

MU

The Current MU driver version is 2.1.3.

- 2.1.3
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.1.2
  - Bug Fixes
    - Fixed issue that MU_GetInstance() is defined but never used.
- 2.1.1
  - Bug Fixes
    - Fixed general interrupt comment typo.
- 2.1.0
  - Improvements
    - Added new enum mu_msg_reg_index_t.
- 2.0.7
  - Bug Fixes
    - Fixed MU_GetInterruptsPending bug that can not get general interrupt status.
- 2.0.6
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.5
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 14.4, 15.5.
- 2.0.4
  - Improvements
    - Improved for the platforms which don't support reset assert interrupt and get the other core power mode.
- 2.0.3
  - Bug fixes
    - MISRA C-2012 issue fixed.
      - Fixed rules, containing: rule-10.3, rule-14.4, rule-15.5.
- 2.0.2

  - Improvements
    - Added support for MIMX8MQx.
- 2.0.1
  - Improvements
    - Added support for MCIMX7Ux_M4.
- 2.0.0
  - Initial version.

PDM

The current PDM driver version is 2.9.1.

- 2.7.4
  - Bug Fixes
    - Fixed the issue that the driver still enters the interrupt after disabling clock.
- 2.9.0
  - Improvements
    - Added feature FSL_FEATURE_PDM_HAS_DECIMATION_FILTER_BYPASS to config CTRL_2[DEC_BYPASS] field.
    - Modify code to make the OSR value is not limited to 16.
- 2.8.1
  - Improvements
    - Added feature FSL_FEATURE_PDM_HAS_NO_DOZEN to handle nonexistent CTRL_1[DOZEN] field.
- 2.8.0
  - Improvements
    - Added feature FSL_FEATURE_PDM_HAS_NO_HWVAD to remove the support of hadware voice activity detector.
    - Added feature FSL_FEATURE_PDM_HAS_NO_FILTER_BUFFER to remove the support of FIR_RDY bitfield in STAT register.
- 2.7.4
  - Bug Fixes
    - Fixed driver can not determine the specific float number of clock divider.
    - Fixed PDM_ValidateSrcClockRate calculates PDM channel in wrong method issue.
- 2.7.3
  - Improvements
    - Added feature FSL_FEATURE_PDM_HAS_NO_VADEF to remove the support of VADEF bitfield in VAD0_STAT register.
- 2.7.2
  - Improvements
    - Added feature FSL_FEATURE_PDM_HAS_NO_MINIMUM_CLKDIV to decide whether the minimum clock frequency division is required.
- 2.7.1
  - Bug Fixes

        * Fixed violations of the MISRA C-2012 rules 8.4, 10.3, 10.1, 10.4, 14.4
- 2.7.0
  - Improvements
    * Added api PDM_EnableHwvadInterruptCallback to support handle hwvad IRQ in PDM driver.
    * Corrected the sample rate configuration for non high quality mode.
    * Added api PDM_SetChannelGain to support adjust the channel gain.
- 2.6.0
  - Improvements
    * Added new features FSL_FEATURE_PDM_HAS_STATUS_LOW_FREQ/FSL_FEATURE_PDM_HAS_DC_OUT_CTRL/FSL_FEATURE_PDM_DC_CTRL_VALUE_FIXED.
- 2.5.0
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 8.4, 16.5, 10.4, 10.3, 10.1, 11.9, 17.7, 10.6, 14.4, 11.8, 11.6.
- 2.4.1
  - Bug Fixes
    * Fixed MDK 66-D warning in pdm driver.
- 2.4.0
  - Improvements
    * Added api PDM_TransferSetChannelConfig/PDM_ReadFifo to support read different width data.
    * Added feature FSL_FEATURE_PDM_HAS_RANGE_CTRL and api PDM_ClearRangeStatus/PDM_GetRangeStatus for range register.
  - Bug Fixes
    * Fixed violation of MISRA C-2012 Rule 14.4, 10.3, 10.4.
- 2.3.0
  - Improvements
    * Enabled envelope/energy voice detect mode by adding apis PDM_SetHwvadInEnvelopeBasedMode/PDM_SetHwvadInEnergyBasedMode.
    * Added feature FSL_FEATURE_PDM_CHANNEL_NUM for different SOC.
- 2.2.1
  - Bug Fixes
    * Fixed violation of MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.6, 10.7, 11.3, 11.8, 14.4, 17.7, 18.4.
    * Added medium quality mode support in function PDM_SetSampleRateConfig.
- 2.2.0
  - Improvements
    * Added api PDM_SetSampleRateConfig to improve user experience and marked api PDM_SetSampleRate as deprecated.
- 2.1.1
  - Improvements
  - Used new SDMA API SDMA_SetDoneConfig instead of SDMA_EnableSwDone for PDM SDMA driver.

- 2.1.0
  - Improvements
    - Added software buffer queue for transactional API.
- 2.0.1
  - Improvements
    - Improved HWVAD feature.
- 2.0.0
  - Initial version.

RDC

The current RDC driver version is 2.2.0.

- 2.2.0
  - New Features
    - Added APIs to get memory region or peripheral access policy for specific domain.
- 2.1.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.6.
- 2.1.0
  - Improvements
    - Enhanced to support memory region larger than 32-bit address.
- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 11.3, 11.8, 17.7.
- 2.0.1
  - Bug Fixes:
    - Added __DSB after new configuration is set to ensure the new configuration takes effect.
- 2.0.0
  - Initial version.

RDC_SEMA42

The current RDC_SEMA42 driver version is 2.0.4.

- 2.0.4
  - Improvements
    - Changed to implement RDC_SEMAPHORE_Lock base on RDC_SEMAPHORE_TryLock.
- 2.0.3
  - Improvements:
    - Supported the RDC_SEMAPHORE_Type structure whose gate registers are defined as an array.
- 2.0.2

  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 10.8, 14.3, 14.4, 18.1.
- 2.0.1
  - Improvements:
    - Added support for the platforms that don't have dedicated RDC_SEMA42 clock gate.
- 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_TransferTxSetConfigSDM-A.
    - Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CH-ANNEL_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.

SDMA

The current SDMA driver version is 2.4.2.

- 2.4.2
  - Bug Fixes
    - Add global variable s_ramScriptLoaded to fix compatibility issues.
- 2.4.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.4, 11.8.
- 2.4.0
  - Improvements
    - Added load ram script automatically according to the peripheral type.
    - Updated sdma ram script to version I.MX7D_4_6.
    - Added api SDMA_GetRamScriptVersion to support to SDMA ram script version.
  - Bug Fixes
    - Removed the SDMA_StartChannelEvents from SDMA_StartTransfer as sdma scheduler will handle it automatically
- 2.3.6
  - Bug Fixes
    - Cleared the SDMAARM_CONFIG_CSM initial value before write value 0 using OR.
- 2.3.5
  - Bug Fixes
    - Added transfer size validation to aovid overflow.
- 2.3.4
  - Bug Fixes
    - Fixed the violation of MISRA C-2012 rule 10.4.
- 2.3.3
  - Improvements
    - Improved sdma driver comments and parameters validation.
- 2.3.2
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.6.
- 2.3.1
  - Improvements
    - Removed clear all channel interrupt status in SDMA_HandleIRQ to avoid the possibility of lossing interrupt.
- 2.3.0
  - Improvements
    - Added peripheral-to-peripheral support in SDMA driver.
    - Added 24bit data width support in sdma driver.
  - Bug Fixes
    - Fixed Coverity issue: left shift may overflow issue.
    - Fixed MISRA2004 issue: the operand of underlying type 'unsigned char' or 'unsigned short' caused the result cast to the underlying type.
    - Fixed violations of MISRA C-2012 rule 10.3, 11.9, 10.4, 17.7, 20.7, 14.4, 11.6, 12.2, 16.4.
- 2.2.1
  - Bug Fixes

        * Fixed MISRA 2004 issue in sdma driver.
- 2.2.0
  - Improvements
    * Added fsl_sdma_script.h to define the sdma script address and firmware.
    * Updated the format of generic register R7 to align with newest firmware.
- 2.1.1
  - Improvements
    * Added SDMA_SetDoneConfig to support hardware/software done configuration.
    * Marked SDMC_EnableSwDone as deprecated.
  - Bug Fixes
    * Fixed logical dead code issue in function SDMA_SetDoneConfig.
- 2.1.0
  - Improvements
    * Added SDMA_SetMultiFifoConfig API to support multi fifo feature.
    * Added SDMA_EnableSwDone API to support software done feature.
    * Added SDMA_LoadScript API to support load script to SDMA program memory.
    * Added SDMA_DumpScript API to support dump script from SDMA program memory.
    * Added SDMA3 IRQ handler.
- 2.0.0
  - Initial version.

SEMA4

The current SEMA4 driver version is 2.0.3.

- 2.0.3
  - Improvements
    * Changed to implement SEMA4_Lock base on SEMA4_TryLock.
- 2.0.2
  - Improvements:
    * Supported the SEMA4_Type structure whose gate registers are defined as an array.
- 2.0.1
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 15.5, 18.1, 18.4.
- 2.0.0
  - Initial version.

TMU

The current TMU driver version is 2.0.0.

- 2.0.0
  - Initial version.
  - This module was first developed on i.MX 8MP to support multi-site temperature monitor.

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.

2 Middleware Change Log

Multicore SDK

The current version of Multicore SDK is 2.16.0

- 2.16.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.13.0
    - eRPC generator (erpcgen) v.1.13.0
    - Multicore Manager (MCMgr) v4.1.5
    - RPMsg-Lite v5.1.2
  - New features:
    - eRPC,erpcgen: Fixing/improving markdown files, GitHub PR #395.
    - eRPC: Fix Python client TCPTransports not being able to close, GitHub PR #390.
    - eRPC,erpcgen: Align switch brackets, GitHub PR #396.
    - eRPC,erpcgen: Remove cstbool library, GitHub PR #403.
    - erpc: Fix zephyr uart transport, GitHub PR #410.
    - erpc: Add BSD-3 license to endianness agnostic files, GitHub PR #417.
    - erpc: UART ZEPHYR Transport stop to work after a few transactions when using USB-CDC resolved, GitHub PR #420.
    - eRPC: Add new Zephyr-related transports (zephyr_uart, zephyr_mbox).
    - eRPC: Add new Zephyr-related examples.
    - RPMsg-Lite: Zephyr-related changes.
    - RPMsg-Lite: Minor Misra corrections.
- 2.15.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.12.0
    - eRPC generator (erpcgen) v.1.12.0
    - Multicore Manager (MCMgr) v4.1.5
    - RPMsg-Lite v5.1.1
  - New features:
    - eRPC: Add dynamic/static option for transport init, GitHub PR #361.
    - eRPC: Fix receive error value for spidev, GitHub PR #363.
    - eRPC: UartTransport::init adaptation to changed driver.
    - eRPC: Fix typo in assert, GitHub PR #371.
    - eRPC,erpcgen: Move enums to enum classes, GitHub PR #379.
    - eRPC: Fixed rpmsg tty transport to work with serial transport, GitHub PR #373.
    - eRPC,erpcgen: Winsock2 support, GitHub PR #365.
    - eRPC,erpcgen: Feature/support multiple clients, GitHub PR #271.
    - eRPC,erpcgen: Feature/buffer head - Framed transport header data stored in MessageBuffer, GitHub PR #378.
    - eRPC,erpcgen: Add experimental Java support.
    - MCMgr: Added notification into MCMGR_EarlyInit and mcmgr_early_init_internal

functions to avoid using uninitialized data in their implementations.
    - RPMsg-Lite: Minor changes in platform and env. layers, minor test code updates.
- 2.14.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.11.0
    - eRPC generator (erpcgen) v.1.11.0
    - Multicore Manager (MCMgr) v4.1.4
    - RPMsg-Lite v5.1.0
  - New features:
    - eRPC: Makefiles update, GitHub PR #301.
    - eRPC: Resolving warnings in Python, GitHub PR #325.
    - eRPC: Python3.8 is not ready for usage of typing.Any type, GitHub PR #325.
    - eRPC: Improved codec function to use reference instead of address, GitHub PR #324.
    - eRPC: Fix NULL check for pending client creation, GitHub PR #341.
    - eRPC: Replace sprintf with snprintf, GitHub PR #343.
    - eRPC: Use MU_SendMsg blocking call in MU transport.
    - eRPC: New LPSPI and LPI2C transport layers.
    - eRPC: Freeing static objects, GitHub PR #353.
    - eRPC: Fixed casting in deinit functions, GitHub PR #354.
    - eRPC: Align LIBUSBSIO.GetNumPorts API use with libusbsio python module v. 2.1.-11.
    - erpcgen: Renamed temp variable to more generic one, GitHub PR #321.
    - erpcgen: Add check that string read is not more than max length, GitHub PR #328.
    - erpcgen: Move to g++ in pytest, GitHub PR #335.
    - erpcgen: Use build=release for make, GitHub PR #334.
    - erpcgen: Removed boost dependency, GitHub PR #346.
    - erpcgen: Mingw support, GitHub PR #344.
    - erpcgen: VS build update, GitHub PR #347.
    - erpcgen: Modified name for common types macro scope, GitHub PR #337.
    - erpcgen: Fixed memcpy for template, GitHub PR #352.
    - eRPC,erpcgen: Change default build target to release + adding artefacts, GitHub PR #334.
    - eRPC,erpcgen: Remove redundant includes, GitHub PR #338.
    - eRPC,erpcgen: Many minor code improvements, GitHub PR #323.
    - MCMgr: Avoid calling tx isr callbacks when respective Messaging Unit Transmit Interrupt Enable flag is not set in the CR/TCR register.
    - MCMgr: Messaging Unit RX and status registers are cleared after the initialization.
    - RPMsg-Lite: Resolved issues in ThreadX env. layer implementation.
    - RPMsg-Lite: Added aarch64 support.
    - RPMsg-Lite: Increased the queue size to (2 * RL_BUFFER_COUNT) to cover zero copy cases.
- 2.13.0_imxrt1180a0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.10.0
    - eRPC generator (erpcgen) v.1.10.0

    - Multicore Manager (MCMgr) v4.1.3
    - RPMsg-Lite v5.0.0
  - New features:
    - MCMgr, RPMsg-Lite: Added porting layers for imxrt1180.
    - MCMgr: mu_isr() updated to avoid calling tx isr callbacks when respective Transmit Interrupt Enable flag is not set in the CR/TCR register.
    - RPMsg-Lite, eRPC: RPMsg_Lite queue size adjusted.
    - eRPC: MU transport layer switched to blocking MU_SendMsg() API use.
- 2.13.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.10.0
    - eRPC generator (erpcgen) v.1.10.0
    - Multicore Manager (MCMgr) v4.1.3
    - RPMsg-Lite v5.0.0
  - New features:
    - eRPC: MUTransport adaptation to new supported SoCs.
    - eRPC: Simplifying CI with installing dependencies using shell script, GitHub PR #267.
    - eRPC: Using event for waiting for sock connection in TCP python server, formatting python code, C specific includes, GitHub PR #269.
    - eRPC: Endianness agnostic update, GitHub PR #276.
    - eRPC: Assertion added for functions which are returning status on freeing memory, GitHub PR #277.
    - eRPC: Fixed closing arbitrator server in unit tests, GitHub PR #293.
    - eRPC: Makefile updated to reflect the correct header names, GitHub PR #295.
    - eRPC: Compare value length to used length() in reading data from message buffer, GitHub PR #297.
    - eRPC: Add TCP_NODELAY option to python, GitHub PR #298.
    - eRPC: Replace EXPECT_TRUE with EXPECT_EQ in unit tests, GitHub PR #318.
    - eRPC: Adapt rpmsg_lite based transports to changed rpmsg_lite_wait_for_link_up() API parameters.
    - eRPC, erpcgen: Better distuingish which file can and cannot by linked by C linker, GitHub PR #266.
    - eRPC, erpcgen: Stop checking if pointer is NULL before sending it to the erpc_free function, GitHub PR #275.
    - eRPC, erpcgen: Changed api to count with more interfaces, GitHub PR #304.
    - erpcgen: Check before reading from heap the buffer boundaries, GitHub PR #287.
    - erpcgen: Several fixes for tests and CI, GitHub PR #289.
    - erpcgen: Refactoring erpcgen code, GitHub PR #302.
    - erpcgen: Fixed assigning const value to enum, GitHub PR #309.
    - erpcgen: Enable runTesttest_enumErrorCode_allDirection, serialize enums as int32 instead of uint32.
    - MCMgr: mcmgr_mu_internal.c code adaptation to new supported SoCs.
    - RPMsg-Lite: Improveed debug check buffers implementation - instead of checking the pointer fits into shared memory check the presence in the VirtIO ring descriptors list.
    - RPMsg-Lite: Timeout parameter added to rpmsg_lite_wait_for_link_up API function.

    - RPMsg-Lite: VRING_SIZE is set based on number of used buffers now (as calculated in vring_init) - updated for all platforms that are not communicating to Linux rpmsg counterpart.
    - RPMsg-Lite: Fixed wrong RL_VRING_OVERHEAD macro comment in platform.h files.
    - RPMsg-Lite: Misra corrections.
- 2.12.0_imx93
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.9.1
    - eRPC generator (erpcgen) v.1.9.1
    - Multicore Manager (MCMgr) v4.1.2
    - RPMsg-Lite v4.0.1
  - New features:
    - RPMsg-Lite: Added porting layers for i.mx93 device.
- 2.12.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.9.1
    - eRPC generator (erpcgen) v.1.9.1
    - Multicore Manager (MCMgr) v4.1.2
    - RPMsg-Lite v4.0.0
  - New features:
    - eRPC: Construct the USB CDC transport, rather than a client, GitHub PR #220.
    - eRPC: Fix premature import of package, causing failure when attempting installation of Python library in a clean environment, GitHub PR #38, #226.
    - eRPC: Improve python detection in make, GitHub PR #225.
    - eRPC: Fix several warnings with deprecated call in pytest, GitHub PR #227.
    - eRPC: Fix freeing union members when only default need be freed, GitHub PR #228.
    - eRPC: Fix making test under Linux, GitHub PR #229.
    - eRPC: Assert costumizing, GitHub PR #148.
    - eRPC: Fix corrupt clientList bug in TransportArbitrator, GitHub PR #199.
    - eRPC: Fix build issue when invoking g++ with -Wno-error=free-nonheap-object, GitHub PR #233.
    - eRPC: Fix inout cases, GitHub PR #237.
    - eRPC: Remove ERPC_PRE_POST_ACTION dependency on return type, GitHub PR #238.
    - eRPC: Adding NULL to ptr when codec function failed, fixing memcpy when fail is present during deserialization, GitHub PR #253.
    - eRPC: MessageBuffer usage improvement, GitHub PR #258.
    - eRPC: Get rid for serial and enum34 dependency (enum34 is in python3 since 3.4 (from 2014)), GitHub PR #247.
    - eRPC: Several MISRA violations addressed.
    - eRPC: Fix timeout for Freertos semaphore, GitHub PR #251.
    - eRPC: Use of rpmsg_lite_wait_for_link_up() in rpmsg_lite based transports, GitHub PR #223.
    - eRPC: Fix codec nullptr dereferencing, GitHub PR #264.

    - erpcgen: Fix two syntax errors in erpcgen Python output related to non-encapsulated unions, improved test for union, GitHub PR #206, #224.
    - erpcgen: Fix serialization of list/binary types, GitHub PR #240.
    - erpcgen: Fix empty list parsing, GitHub PR #72.
    - erpcgen: Fix templates for malloc errors, GitHub PR #110.
    - erpcgen: Get rid of encapsulated union declarations in global scale, improve enum usage in unions, GitHub PR #249, #250.
    - erpcgen: Fix compile error:UniqueIdChecker.cpp:156:104:'sort' was not declared, GitHub PR #265.
    - MCMgr: Update mcmgr_stop_core_internal() implementations to set core state to kMCMGR_ResetCoreState.
    - RPMsg-Lite: Introduce new rpmsg_lite_wait_for_link_up() API function - this allows to avoid using busy loops in rtos environments, GitHub PR #21.
    - RPMsg-Lite: Adjust rpmsg_lite_is_link_up() to return RL_TRUE/RL_FALSE.
- 2.11.1
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.9.0
    - eRPC generator (erpcgen) v.1.9.0
    - Multicore Manager (MCMgr) v4.1.1
    - RPMsg-Lite v3.2.1
  - New features:
    - RPMsg-Lite: Add support for custom shared memory arangement per the RPMsg_Lite instance.
- 2.11.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.9.0
    - eRPC generator (erpcgen) v.1.9.0
    - Multicore Manager (MCMgr) v4.1.1
    - RPMsg-Lite v3.2.0
  - New features:
    - eRPC: Improving template usage, GitHub PR #153.
    - eRPC: run_clang_format.py cleanup, GitHub PR #177.
    - eRPC: Build TCP transport setup code into liberpc, GitHub PR #179.
    - eRPC: Fix multiple definitions of g_client error, GitHub PR #180.
    - eRPC: Fix memset past end of buffer in erpc_setup_mbf_static.cpp, GitHub PR #184.
    - eRPC: Fix deprecated error with newer pytest version, GitHub PR #203.
    - eRPC: Allow used LIBUSBSIO device index being specified from the Python command line argument.
    - eRPC, erpcgen: Static allocation support and usage of rpmsg static FreeRTOSs related APi, GitHub PR #168, #169.
    - erpcgen: Remove redundant module imports in erpcgen, GitHub PR #196.
    - RPMsg-Lite: Improve static allocations - allow OS-specific objects being allocated statically, GitHub PR #14.
    - RPMsg-Lite: Minor Misra and typo corrections, GitHub PR #19, #20.
- 2.10.0

  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.8.1
    - eRPC generator (erpcgen) v.1.8.1
    - Multicore Manager (MCMgr) v4.1.1
    - RPMsg-Lite v3.1.2
  - New features:
    - eRPC: Fix misra erpc c, GitHub PR #158.
    - eRPC: Allow conditional compilation of message_loggers and pre_post_action.
    - eRPC: New i2c_slave_transport trasnport introduced.
    - eRPC: (D)SPI slave transports updated to avoid busy loops in rtos environments.
    - erpcgen: Re-implement EnumMember::hasValue(), GitHub PR #159.
    - erpcgen: Fixing several misra issues in shim code, erpcgen and unit tests updated, GitHub PR #156.
    - erpcgen: Fix bison file, GitHub PR #156.
    - RPMsg-Lite: Fixed incorrect description of the rpmsg_lite_get_endpoint_from_addr function.
    - RPMsg-Lite: Updated RL_BUFFER_COUNT documentation.
    - RPMsg-Lite: env_print macro adjusted to address MISRA 21.6 rule in MCUXpressoSDK projects.
- 2.9.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.8.0
    - eRPC generator (erpcgen) v.1.8.0
    - Multicore Manager (MCMgr) v4.1.1
    - RPMsg-Lite v3.1.1
  - New features:
    - eRPC: Support win32 thread, GitHub PR #108.
    - eRPC: Add mbed support for malloc() and free(), GitHub PR #92.
    - eRPC: Update makefile.
    - eRPC: Fixed warnings and error with using MessageLoggers, GitHub PR #127.
    - eRPC: Extend error msg for python server service handle function, GitHub PR #132.
    - eRPC: Update CMSIS UART transport layer to avoid busy loops in rtos environments, introduce semaphores.
    - eRPC: Introduced pre and post callbacks for eRPC call, GitHub PR #131.
    - eRPC: Introduced new USB CDC transport.
    - eRPC: Introduced new Linux spidev-based transport.
    - eRPC: SPI transport update to allow usage without handshaking GPIO.
    - eRPC: Native *WIN32 erpc serial transport and threading.*
    - *eRPC: Arbitrator deadlock fix, TCP transport updated, TCP setup functions introduced, GitHub PR #121.*
    - *eRPC: Update of matrix_multiply.py example: Add –serial and –baud argument, GitHub PR #137.*
    - *eRPC: Added formatting extension for VSC, GitHub PR #134.*
    - *eRPC: Update of .clang-format, GitHub PR #140.*
    - *eRPC: Update of erpc_framed_transport.cpp: return error if received message has zero*

length, GitHub PR #141.
    - *eRPC, erpcgen: Fixed error messages produced by -Wall -Wextra -Wshadow -pedantic-errors compiler flags, GitHub PR #136, #139.*
    - *eRPC, erpcgen: Core re-formatted using Clang version 10.*
    - *erpcgen: Enable deallocation in server shim code when callback/function pointer used as out parameter in IDL.*
    - *erpcgen: Removed '$' character from generated symbol name in '*$union' suffix, GitHub PR #103.
    - erpcgen: Resolved mismatch between C++ and Python for callback index type, GitHub PR #111.
    - erpcgen: Python generator improvements, GitHub PR #100, #118.
    - erpcgen: Fixed error messages produced by -Wall -Wextra -Wshadow -pedantic-errors compiler flags, GitHub PR #136.
    - erpcgen: Introduce ustring type for unsigned char and force cast to char∗, GitHub PR #125.
    - RPMsg-Lite: Introduced RL_ALLOW_CONSUMED_BUFFERS_NOTIFICATION config option to allow opposite side notification sending each time received buffers are consumed and put into the queue of available buffers.
    - RPMsg-Lite: Added environment layers for Threadx.
- 2.8.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.4
    - eRPC generator (erpcgen) v.1.7.4
    - Multicore Manager (MCMgr) v4.1.0
    - RPMsg-Lite v3.1.0
  - New features:
    - eRPC: Unit test code updated to handle service add and remove operations.
    - eRPC: Several MISRA issues in rpmsg-based transports addressed.
    - eRPC: Support MU transport unit testing.
    - eRPC: Adding mbed os support.
    - eRPC: Fixed Linux/TCP acceptance tests in release target.
    - eRPC: Minor documentation updates, code formatting.
    - erpcgen: Whitespace removed from C common header template.
    - RPMsg-Lite: MISRA C-2012 violations fixed (7.4).
    - RPMsg-Lite: Fix missing lock in rpmsg_lite_rx_callback() for QNX env.
    - RPMsg-Lite: Correction of rpmsg_lite_instance structure members description.
    - RPMsg-Lite: Address -Waddress-of-packed-member warnings in GCC9.
    - RPMsg-Lite: Clang update to v10.0.0, code re-formatted.
- 2.7.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.3
    - eRPC generator (erpcgen) v.1.7.3
    - Multicore Manager (MCMgr) v4.1.0
    - RPMsg-Lite v3.0.0
  - New features:

    - eRPC: Improved the test_callbacks logic to be more understandable and to allow requested callback execution on the server side.
    - eRPC: TransportArbitrator::prepareClientReceive modified to avoid incorrect return value type.
    - eRPC: The ClientManager and the ArbitratedClientManager updated to avoid performing client requests when the previous serialization phase fails.
    - erpcgen: Generate the shim code for destroy of statically allocated services.
    - MCMgr: Code adjustments to address MISRA C-2012 Rules
    - RPMsg-Lite: MISRA C-2012 violations fixed, incl. data types consolidation.
    - RPMsg-Lite: Code formatted
- 2.6.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.2
    - eRPC generator (erpcgen) v.1.7.2
    - Multicore Manager (MCMgr) v4.0.3
    - RPMsg-Lite v2.2.0
  - New features:
    - eRPC: Improved support of const types.
    - eRPC: Fixed Mac build.
    - eRPC: Fixed serializing python list.
    - eRPC: Documentation update.
    - eRPC: Add missing doxygen comments for transports.
    - RPMsg-Lite: Added configuration macro RL_DEBUG_CHECK_BUFFERS.
    - RPMsg-Lite: Several MISRA violations fixed.
    - RPMsg-Lite: Added environment layers for QNX and Zephyr.
    - RPMsg-Lite: Allow environment context required for some environments (controlled by the RL_USE_ENVIRONMENT_CONTEXT configuration macro).
    - RPMsg-Lite: Data types consolidation.
    - MCMgr: Documentation updated to describe handshaking in a graphic form.
    - MCMgr: Minor code adjustments based on static analysis tool findings
- 2.5.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.1
    - eRPC generator (erpcgen) v.1.7.1
    - Multicore Manager (MCMgr) v4.0.2
    - RPMsg-Lite v2.0.2
  - New features:
    - RPMsg-Lite, MCMgr: Align porting layers to the updated MCUXpressoSDK feature files.
    - eRPC: Fixed semaphore in static message buffer factory.
    - erpcgen: Fixed MU received error flag.
    - erpcgen: Fixed tcp transport.
- 2.4.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.7.0

        - eRPC generator (erpcgen) v.1.7.0
        - Multicore Manager (MCMgr) v4.0.1
        - RPMsg-Lite v2.0.1
    - New features:
        - eRPC: Improved code size of generated code.
        - eRPC: Generating crc value is optional.
        - eRPC: Fixed CMSIS Uart driver. Removed dependency on KSDK.
        - eRPC: List names are based on their types. Names are more deterministic.
        - eRPC: Service objects are as a default created as global static objects.
        - eRPC: Added missing doxygen comments.
        - eRPC: Forbid users use reserved words.
        - eRPC: Removed outByref for function parameters.
        - eRPC: Added support for 64bit numbers.
        - eRPC: Added support of program language specific annotations.
        - eRPC: Optimized code style of callback functions.
        - RPMsg-Lite: New API rpmsg_queue_get_current_size()
        - RPMsg-Lite: Fixed bug in interrupt handling for lpc5411x, lpc5410x
        - RPMsg-Lite: Code adjustments based on static analysis tool findings
- 2.3.1
    - Multicore SDK component versions:
        - embedded Remote Procedure Call (eRPC) v1.6.0
        - eRPC generator (erpcgen) v.1.6.0
        - Multicore Manager (MCMgr) v4.0.0
        - RPMsg-Lite v1.2.0
    - New features:
        - eRPC: Improved code size of generated code.
        - eRPC: Improved eRPC nested calls.
        - eRPC: Improved eRPC list length variable serialization.
        - eRPC: Added @nullable support for scalar types.
        - MCMgr: Added new MCMGR_TriggerEventForce() API.
- 2.3.0
    - Multicore SDK component versions:
        - embedded Remote Procedure Call (eRPC) v1.5.0
        - eRPC generator (erpcgen) v.1.5.0
        - Multicore Manager (MCMgr) v3.0.0
        - RPMsg-Lite v1.2.0
    - New features:
        - eRPC: Added support for unions type non-wrapped by structure.
        - eRPC: Added callbacks support.
        - eRPC: Added support @external annotation for functions.
        - eRPC: Added support @name annotation.
        - eRPC: Added Messaging Unit transport layer.
        - eRPC: Added RPMSG Lite RTOS TTY transport layer.
        - eRPC: Added version verification and IDL version verification between eRPC code and eRPC generated shim code.

    - eRPC: Added support of shared memory pointer.
    - eRPC: Added annotation to forbid generating const keyword for function parameters.
    - eRPC: Added python matrix multiply example.
    - eRPC: Added nested call support.
    - eRPC: Added struct member "byref" option support.
    - eRPC: Added support of forward declarations of structures
    - eRPC: Added Python RPMsg Multiendpoint kernel module support
    - eRPC: Added eRPC sniffer tool
    - MCMgr: Unused API removed
    - MCMgr: Added the ability for remote core monitoring and event handling
    - RPMsg-Lite: Several source files renamed to avoid conflicts with other middleware sw components
    - RPMsg-Lite: Added the ability to use Multicore Manager (MCMGR) as the IPC interrupts router
- 2.2.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.4.0
    - eRPC generator (erpcgen) v.1.4.0
    - Multicore Manager (MCMgr) v2.0.1
    - RPMsg-Lite v1.1.0
  - New features:
    - eRPC: win_flex_bison.zip for windows updated.
    - eRPC: Use one codec (instead of inCodec outCodec).
    - eRPC: New RPMsg-Lite Zero Copy (RPMsgZC) transport layer.
    - MCMgr: code updated to be Misra compliant.
    - RPMsg-Lite: Added macros for packed structures (compiler.h).
    - RPMsg-Lite: Improved interrupt handling in platform layer.
    - RPMsg-Lite: Changed RL_BUFFER_SIZE definition.
    - RPMsg-Lite: Fix of double initialization of vring shared data structure.
    - RPMsg-Lite: Support for the multi-instance.
- 2.1.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.3.0
    - eRPC generator (erpcgen) v.1.3.0
  - New features:
    - eRPC: New annotation types introduced (@length, @max_length, ...).
    - eRPC: Support for running both erpc client and erpc server on one side.
    - eRPC: New transport layers for (LP)UART, (D)SPI.
    - eRPC: Error handling support.
- 2.0.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.2.0
    - eRPC generator (erpcgen) v.1.2.0
    - Multicore Manager (MCMgr) v2.0.0
    - RPMsg-Lite v1.0.0

  - New features:
    - Multicore SDK support for lpcxpresso54114 board added.
    - RPMsg component of the Open-AMP framework re-implemented and the RPMsg-Lite version introduced.
    - eRPC source directory organization changed.
    - Many eRPC improvements.
- 1.1.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.1.0
    - Multicore Manager (MCMgr) v1.1.0
    - Open-AMP / RPMsg based on SHA1 ID 44b5f3c0a6458f3cf80 rev01
  - New features:
    - Multicore SDK 1.1.0 ported to KSDK 2.0.0.
    - Python support added into eRPC.
- 1.0.0
  - Multicore SDK component versions:
    - embedded Remote Procedure Call (eRPC) v1.0.0
    - Multicore Manager (MCMgr) v1.0.0
    - Open-AMP / RPMsg based on SHA1 ID 44b5f3c0a6458f3cf80 rev00

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.

.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.

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