MCUXpresso SDK Release Notes Supporting evkmimxrt1170

Change Logs

NXP Semiconductors

Contents

Driver Change Log

Title

Page No.

Title

Page No.

Title

Page No.

CANopen Change Log

Title

Page No.

Component Change Log

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.5.5.

- 2.5.5
  - Improvements
    - Move s_clockSourceName array to c from header.
- 2.5.4
  - Improvements
    - Toggle hold_ring_off during arm pll initialization.
  - Bug Fixes
    - Fixed an issue that in CLOCK_InitSysPll3() some reserved bits may be cleared.
- 2.5.3
  - Bug Fixes
    - Fixed bug in CLOCK_GetPllFreq() the formula is not general.
    - Fixed bug in CLOCK_InitSysPll1() function.
- 2.5.2
  - Bug Fixes
    - Fixed bug in XBARA_CLOCKS macro defie.
- 2.5.1
  - Bug Fixes
    - Fixed bug in CLOCK_InitArmPll() function.
    - Fixed bug clock root divider set to cut off at 255.
- 2.5.0
  - New Features
    - Added CLOCK_DeinitPfd() function to clock gate selected PLL's pfd.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 5.6.
    - Fixed bug in XBARA_CLOCKS macro define.
- 2.4.0
  - New features
    - Added CLOCK_IsUsb1PfdEnabled and CLOCK_IsSysPfdEnabled to get the clock source status.
  - Improvements
    - In CLOCK_InitPfd() function, check fractional divide value firstly, if register's value is equal to value to set, return directly.
  - Bug Fixes
    - Fixed Doxygen warnings.
    - Fixed violations of MISRA C-2012 rule 10.3, 10.1, 10.4, 14.4, 16.6, 16.1, 8.8.
- 2.3.0
  - Improvements
    - Added APIs for CCM functional blocks: OSCPLL, CLOCK_ROOT, LPCG.

        - Added APIs for Unassigned Mode, Domain Mode, CPU Low Power Mode, SetPoint Mode.
        - Added APIs for control mode access and white list access.
        - Added APIs for OSCPLL, LPCG to set current accessing domain's clock depend level.
      - Added APIS for Fractional PLLs during GPC mode.
      - Added kCLOCK_SysPll3Div2 case in CLOCK_GetFreq() function.
- 2.2.0
  - Improvements
    - Updated PLL initialization API for those PLLs support spread spectrum
    - Added APIs for PLL
      - check/set PLL bypass
      - check if PLL is enabled
- 2.1.10
  - Improvements
    - Defined SDK_DEVICE_MAXIMUM_CPU_CLOCK_FREQUENCY as 400000000UL for m4 core.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules, including 10.3, 16.1, 16.4, 9.3, 10.4, 10.8, 5.6, 11.1, 14.3, 10.7, 14.4, 12.2, 8.4.
- 2.1.9
  - Bug Fixes
    - Aligned behavor of clock root divider setting and geting
- 2.1.8
  - Bug Fixes
    - Fixed CLOCK_SetRootClockMux changes not mux only
- 2.1.7
  - New Features
    - Supported ARM PLL initialization with frequency.
- 2.1.6
  - Improvements
    - Added puf related enum variable in clock driver.
- 2.1.5
  - Bug Fixes
    - Fixed CLOCK_GetRootClockFreq does not get correct OSC RC 16M frequency.
- 2.1.4
  - Bug Fixes
    - rt1170: fix set clock root divider issue
    - rt1170: add __xSB() after changing clock related registers
- 2.1.3
  - Improvements
    - Aligned naming rule for enum variable _clock_pll with previous RT1xxx devices.
- 2.1.2
  - Improvements
    - Added a new enumerator to define the mux value of clock source in clock roots.

- 2.1.1
  - Bug Fixes
    - Fixed AV Pll frequency calculation
- 2.1.0
  - New features
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.0.0
  - initial version.

IOMUXC

The current IOMUXC driver version is 2.0.1.

- 2.0.1
  - Doxygen improvement.
- 2.0.0
  - initial version.

ANATOP_AI

The current anatop_ai driver version is 2.0.0.

- 2.0.0
  - initial version.

SOC_MIPI_CSI2RX

The current SOC_MIPI_CSI2RX driver version is 2.0.2.

- 2.0.2
  - Updated for new header file.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA-C 2012 10.8 issue.
- 2.0.0
  - initial version.

SRC

The current SRC driver version is 2.1.1.

- 2.1.1
  - Bug Fixes

        * Fixed Doxygen warnings.
- 2.1.0
  - Improvements
    * Provided more flexible APIs to set slice authentication.
- 2.0.1
  - Bug Fixes
    * Fixed the violation of MISRA C-2012 rule 10.4.
- 2.0.0
  - Initial version.

GPC

The current GPC driver version is 2.3.1.

- 2.3.1
  - Bug Fixes
    * Fixed the violation of MISRA C-2012 rule 5.8.
- 2.3.0
  - Bug Fixes
    * Fixed wrong offset value of DCDC_UP_CTRL register.
  - New Features
    * Added GPC_STBY_ForceCoreRequestStandbyMode() function ti force core to enter standby mode.
- 2.2.0
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.8.
    * Fixed violations of MISRA C-2012 rule 8.6 by removing the declaration of GPC_SP_-GetResponseCount() function.
- 2.1.1
  - Bug Fixes
    * Fixed Doxygen warnings.
- 2.1.0
  - Improvements
    * Removed status related APIs based on the updates of header file.
- 2.0.0
  - Initial version.

PGMC

The current PGMC driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    * Fixed bug in PGMC_PPC_TriggerPMICStandbySoftMode() function.

- 2.1.1
  - Bug Fixes
    - Fixed Doxygen warnings.
- 2.1.0
  - Improvements
    - Updated PGMC driver based on the updates of header file.
- 2.0.0
  - Initial version.

PMU

The current PMU driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    - Updated PMU_StaticEnablePllLdo() with disabling LDO current limit after LDO is stable to minimize ARM PLL jitter in cold temperature.
- 2.1.1
  - Bug Fixes
    - Fixed bugs in FBB configuration.
    - Updated delay value from 1us to 100us in PMU_StaticEnablePllLdo() function.
- 2.1.0
  - Improvements
    - Updated the PMU driver based on the new header file.
    - Defined the macro to separate different scenes that some devices may do not support FBB.
    - Fixed Doxygen warnings.
    - Fixed violations of MISRA C-2012 rule 14.3.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 13.1, rule 10.1, rule 10.4, and rule 14.3.
- 2.0.0
  - Initial version.

ROMAPI

The current ROMAPI driver version is 1.1.1.

- 1.1.1
  - Improvements
    - Update the comments of "clear cache" function.
- 1.1.0
  - New features
    - Support B0 silicon
- 1.0.0
  - initial version.

SOC_FLEXRAM_ALLOCATE

The current SOC_FLEXRAM_ALLOCATE driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.4.
- 2.0.1
  - Bug Fixes
    - Fixed wrong GPR setting for re-allocate OCRAM/DTCM/ITCM.
- 2.0.0
  - initial version.

DCDC

The current DCDC driver version is 2.1.2.

- 2.1.2
  - Improvements
    - The DCDC_GetInstance() function is only available when FSL_SDK_DISABLE_DRIVER_CLOCK_CONTROL is set to 0.
- 2.1.1
  - Bug Fixes
    - Fixed Doxygen warnings.
- 2.1.0
  - Improvements
    - Updated DCDC_BootIntoDCM() function.
    - Based on the updates of header file, updated dcdc driver.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.3, rule 10.7, and rule 12.2.
- 2.0.0
  - Initial version.

NIC301

The current NIC301 driver version is 2.0.1.

- 2.0.1
  - Bug Fixes.
    - Fixed the repeat of offset addition in this file API.
- 2.0.0
  - 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.

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.

ACMP

The current ACMP driver version is 2.3.0.

- 2.3.0
  - Improvements
    - Expose C0 register FILTER_CNT bitfield and FPR bitfield to the user.
- 2.2.0
  - Improvements
    - Updated feature macros for roundrobin mode, window mode, filter mode, and 3V domain removes.
- 2.1.0
  - New Feature

        * Supported the plateforms which don't have hysteresis mode.
- 2.0.6
  - Bug Fixes
    * Fixed the wrong comments, the DAC value should range from 0 to 255.
- 2.0.5
  - Bug Fixes
    * Fixed the out-of-bounds error of Coverity caused by missing an assert sentence to avoid the return value of ACMP_GetInstance() exceeding the array bounds.
    * Fixed the violations of MISRA C-2012 rules:
      · Rule 10.1, 14.4, 16.4, 17.7.
- 2.0.4
  - Bug Fixes
    * Avoided changing w1c bit in ACMP_SetRoundRobinPreState().
- 2.0.3
  - New Features
    * Added feature functions for usage of different power domains(1.8 V and 3 V). These functions are first enabled in ULP1. They are about:
      · ACMP_EnableLinkToDAC()
      · ACMP_SetDiscreteModeConfig()
      · ACMP_GetDefaultDiscreteModeConfig()
- 2.0.2
  - Other Changes
    * Changed coding style of peripheral base address from "s_acmpBases" to "s_acmpBase".
- 2.0.1
  - Bug Fixes
    * Fixed bug regarding the function "ACMP_SetRoundRobinConfig". It will not continue execution but returns directly after disabling round robin mode.

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.

ASRC

The current ASRC driver version is 2.1.3.

- 2.1.3
  - Bug Fixes
    - Fixed function did not match the specified channel pair issue.
- 2.1.2
  - Improvements

    - Correct feature name in source file by changing FSL_FEATURE_ASRC_PARAMETER_REGISTER_NAME_ASPRM to FSL_FEATURE_ASRC_PARAMETER_REGISTER_NAME_ASRPM.
    - Removed the asrc_clock_source_t from driver header file, as SOC header file will provide detail definition.
  - Bug Fixes
    - Fixed the ASRC_SetChannelPairConfig/ASRC_ChannelPairEnable functions missing functionality when using channel pair B/C.
    - Fixed violations of the MISRA C-2012 rules 10.7.
- 2.1.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.110.4, 12.2.
- 2.1.0
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 14.4, 10.1, 17.7, 11.9, 8.6, 12.2, 11.6.
- 2.0.1
  - Improvements
    - Added feature macro FSL_FEATURE_ASRC_PARAMETER_REGISTER_NAME_ASPRM for ASRC parameter register.
  - Bug Fixes
    - Fixed the unused build warning in asrc edma driver.
- 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.

CSI

The current CSI driver version is 2.1.5.

- 2.1.5
  - Improvements
    - Updated for new CSI register and macro names.
- 2.1.4
  - Improvements
    - Added memory address conversion to support buffers which could only be accessed using alias address by non-core masters.
- 2.1.3

  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.1.2
  - Improvements
    - Supported new CSI_Type register naming.
- 2.1.1
  - Bug Fixes
    - Fixed IAR build warning Pa082.
    - Fixed violations of the MISRA C-2012 rules 8.4, 10.1, 10.3, 10.4, 10.6, 11.6, 14.4, 17.7.
- 2.1.0
  - New Features
    - Added 16-bit and 24-bit data bus support.
  - Bug Fixes:
    - Fixed the bug that CSI writes to wrong buffer when empty buffer not submitted in time.
- 2.0.3
  - Bug Fixes
    - Fixed wrong circular queue delta calculation.
    - Fixed double buffering capture issue where, when the transfer is ongoing and the device has empty buffer slot, the function CSI_TransferSubmitEmptyBuffer sets the empty buffer to CSI device.
- 2.0.2
  - New Features
    - Added fragment mode support.
- 2.0.1
  - Improvements
    - Switched DMA output buffer at the first data after each VSYNC. It originally happened when the DMA transfer was done.
- 2.0.0
  - Initial version.

DAC12

The current DAC12 driver version is 2.1.1.

- 2.1.1
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.1.0
  - Improvements
    - Defined the macro "FSL_FEATURE_HAS_NO_ITRM_REGISTER" to distinguish different scenes that ITRM register may not equipped one some devices.
- 2.0.1
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rules:

        - Rule 10.8, 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.

ELCDIF

The current ELCDIF driver version is 2.0.7.

- 2.0.7
  - Bug Fixes
    - Fixed faulty operation of CTRL1 in ELCDIF_RgbModeSetPixelFormat.
- 2.0.6
  - Bug Fixes
    - Fixed bug in ELCDIF_RgbModeStop that the API shall return until RUN bit is cleared, so that the RGB mode is properly stopped.
- 2.0.5
  - Bug Fixes
    - Fixed the violations of MISRA 2012 advisory rules.
- 2.0.4
  - Improvements

        * Increase outstanding transactions for better performance.
        * Added memory address conversion to support buffers which could only be accessed using alias address by non-core masters.
- 2.0.3
  - Improvements
    * Supported the platforms which don't have PXP handshake feature.
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 3.1, 8.4, 10.1, 10.6, 10.7, 10.8, 14.4, 17.7
    * Removed hardcode delay in function ELCDIF_Reset.
- 2.0.1
  - Improvements
    * Added the function ELCDIF_RgbModeSetPixelFormat.
- 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.

ENET_QOS

The current ENET_QOS driver version is 2.6.4.

- 2.6.4
  - Improvements
    * ENET_QOS_SetMII returns success or failure status now (related to i.MX RT1170 errata ERR050539 - ENET_QOS doesn't support RMII 10Mbps mode).
  - Bug Fixes
    * Fixed the MISRA C-2012 issue rule 14.3.
- 2.6.3
  - Bug Fixes
    * Fixed the issue that ENET_QOS_GetRxFrame, ENET_QOS_ReadFrame and ENET_QOS_DropFrame did not properly restart the receiving once it stopped.
- 2.6.2
  - 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.
- 2.6.1
  - Bug Fixes
    * Fixed the issue that ENET_QOS_ReadFrame doesn't check timestamp available bit before check the context BD bit, it makes software update extra BD. If DMA receives new frame to this BD before software update, software will lose this frame.

- 2.6.0
  - New features
    - Added hardware checksum acceleration support.
- 2.5.3
  - Bug Fixes
    - Fixed the MISRA issue rule 14.3, 5.3.
- 2.5.2
  - Bug Fixes
    - Fixed the issue that ENET_QOS_Init reset the MDIO setting of ENET_QOS_SetSMI.
- 2.5.1
  - Improvements
    - Supported RMII mode.
- 2.5.0
  - Improvements
    - Added MDIO access wrapper APIs for ease of use.
- 2.4.1
  - Improvements
    - Supported cache control.
    - Supported BD address convertion to system address.
    - Make driver aarch64 compatible
  - Bug Fixes
    - Fixed the issue that driver internal interface ENET_QOS_DropFrame drops all frames in whole BD ring rather than one frame as design. Impact case: 1. Rx drop occurs in zero copy Rx API ENET_QOS_GetRxFrame. 2. Call ENET_QOS_ReadFrame with data pointer is NULL, driver will drop all Rx frames.
- 2.4.0
  - New features
    - Added MDIO IEEE802.3 Clause 45 access support.
    - Added get statistics API to get some statistical data in transfer.
    - Added new APIs to support zero copy Rx.
    - Fixed the MISRA issue rule 8.4, 8.6.
- 2.3.0
  - Improvements
    - 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.
  - Bug Fixes
    - Updated txDirtyRing maintenance in reclaim and send frame process, allow txDirtyRing to be overwritten.
    - Disabled carrier sensing in full duplex mode configuration in ethernet initialization
    - Fixed 1588 sub-second calculate issue.
- 2.2.2
  - Bug Fixes
    - Fixed the issue that ENET_QOS_SetupTxDescriptor didn't handle the DMA access address mapping for SoCs have feature FSL_FEATURE_MEMORY_HAS_ADDRES-

S_OFFSET.
        * Fixed MISRA 2012 violations detected in examples build.
- 2.2.1
  - Bug Fixes
    * Fixed MISRA 2012 violations, fixed doxygen warning.
    * Fixed the issue that cache invalidate to invalid converted memory address in ENET_QOS_ReadFrame for SoCs have feature FSL_FEATURE_MEMORY_HAS_ADDRESS_OFFSET.
- 2.2.0
  - Removed the ptp time data ring management, below structures and APIs are removed:
    * structure enet_qos_ptp_time_data_t
    * structure enet_qos_ptp_time_data_ring_t
    * API ENET_QOS_GetRxFrameTime
    * API ENET_QOS_GetTxFrameTime
  - Added API for GCL list read and AVB configuration
    * ENET_QOS_EstReadGcl
    * ENET_QOS_AVBConfigure
  - Improved driver for PTP system time configuration, timestamp read.
  - Added IRQ lock and memory barrier instruction for descriptor operation.
  - Fixed MISRA 2012 violations
- 2.1.1
  - Bug Fixes
    * Fixed the bug that data pointer is not converted to local memory address in the call to ENET_QOS_Ptp1588ParseFrame.
- 2.1.0
  - New feature
    * Update driver to support feature FSL_FEATURE_MEMORY_HAS_ADDRESS_OFFSET which convert buffer address to visible address for DMA.
    * Require user to provide implementation for ENET_QOS_SetSYSControl API, which set the PHY interface and enable clock generation for IP.
- 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.

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.

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_FEA-

TURE_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_FEATURE_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 FLEXSPI_TransferHandleIRQ.
    - Updated eDMA API FLEXSPI_TransferEDMA.
- 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.

IEE

The current IEE driver version is 2.1.1.

- 2.1.1
  - Fixed MISRA issues.
- 2.1.0
  - Add region lock function IEE_LockRegionConfig() and driver clock control.
- 2.0.0
  - Initial version.

IEE_APC

The current IEE_APC driver version is 2.0.1.

- 2.0.1
  - Fixed MISRA issues.
- 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.

KEYMGR

The current Key Manager driver version is 2.0.2.

- 2.0.2
  - Fix MISRA-2012 issues.
- 2.0.1
  - Fix MISRA-2012 issues.

- 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.

LCDIFv2

Current LCDIFv2 driver version is 2.3.3

- 2.3.3
  - Other Changes
    - Removed PDI_PARA register operation due to IP change.
- 2.3.2
  - Bug Fixes
    - Fixed the violations of MISRA 2012 advisory rules.
- 2.3.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.4.
- 2.3.0
  - New Features:
    - Added API to calculate global alpha based on desired blended alpha.
- 2.2.3
  - Improvements
    - Added memory address conversion to support buffers which could only be accessed using alias address by non-core masters.
- 2.2.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.2, 10.4, 10.6, 12.2.
- 2.2.1
  - Improvements
    - Updated for the new LCDIFV2_Type structure.
- 2.2.0
  - Bug Fixes
    - Fixed LCDIFV2_GetPorterDuffConfig issue that does not set color mode correctly.
  - Other Changes

        - Removed the store functions.
- 2.1.1
  - Bug Fixes
    - Fixed the issue that LCDIFV2_SetLut could not access the last index.
- 2.1.0
  - New Features:
    - Added function to get Porter Duff configuration.
- 2.0.1
  - Bug Fixes
    - Fixed the issue that register value not reset by LCDIFV2_Deinit and LCDIFV2_Reset.
- 2.0.0
  - Initial version.

LPADC

The current LPADC driver version is 2.8.4.

- 2.8.4
  - Bug Fixes
    - Remove function 'LPADC_SetOffsetValue' assert statement, this statement may cause runtime errors in existing code.
- 2.8.3
  - Bug Fixes
    - Fixed SDK lpadc driver examples compile issue, move condition 'commandId < ADC_CV_COUNT' to a more appropriate location.
- 2.8.2
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rule 18.1, 10.3, 10.1 and 10.4.
- 2.8.1
  - Bug Fixes
    - Fixed LPADC sample mode enum name mistake.
- 2.8.0
  - Improvements
    - Release peripheral from reset if necessary in init function.
  - Bug Fixes
    - Fixed function LPADC_GetConvResult() issue.
    - Fixed function LPADC_SetConvCommandConfig() bugs.
- 2.7.2
  - Improvements
    - Use feature macros instead of header file macros.
  - Bug Fixes
    - Fixed the violations of MISRA C-2012 rule 10.1, 10.3, 10.4 and 14.3.
- 2.7.1
  - Improvements

        * Corrected descriptions of several functions.
        * Improved function LPADC_GetOffsetValue and LPADC_SetOffsetValue.
        * Revert changes of feature macros for lpadc.
        * Use feature macros instead of header file macros.
    - Bug Fixes
        * Fixed the violations of MISRA C-2012 rule 10.8.
        * Fixed the violations of MISRA C-2012 rule 10.1, 10.3, 10.4 and 14.3.
- 2.7.0
    - Improvements
        * Added supports of CFG2 register.
        * Removed some useless macros.
- 2.6.2
    - Bug Fixes
        * Fixed the violations of MISRA C-2012 rules.
        * Fixed LPADC driver code compile error issue.
- 2.6.1
    - Improvements
        * Updated the use of macros in the driver code.
- 2.6.0
    - Improvements
        * Added the API LPADC_SetOffset12BitValue() to configure 12bit ADC conversion offset trim value manually.
        * Added the API LPADC_SetOffset16BitValue() to configure 16bit ADC conversion offset trim value manually.
        * Added API to set offset calibration mode.
        * Added configuration of alternate channel.
        * Updated auto calibration API and added calibration value conversion API.
    - New feature
        * Added API LPADC_EnableHardwareTriggerCommandSelection() to enable trigger commands controlled by ADC_ETC.
        * Updated LPADC_DoAutoCalibration() to allow doing something else before the ADC inititialization to be totally complete. Enhance initialization duration time of the ADC.
        * Added two new APIs to get/set calibration value.
- 2.5.2
    - Improvements
        * Added while loop, LPADC_GetConvResult() will return only when the FIFO will not be empty.
- 2.5.1
    - Bug Fixes
        * Fixed some typos in Lpadc driver comments.
- 2.5.0
    - Improvements
        * Added missing items to enable trigger interrupts.
- 2.4.0
    - New features

        * Added APIs to get/clear trigger status flags.
- 2.3.0
  - Improvements
    * Removed LPADC_MeasureTemperature() function for the LPADC supports different temperature sensor calculation equations.
- 2.2.1
  - Improvements
    * Optimized LPADC_MeasureTemperature() function to support the specific series with flash solidified calibration value.
    * Clean doxygen warnings.
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.3, rule 10.8 and rule 17.7.
- 2.2.0
  - New Feature
    * Added API LPADC_MeasureTemperature() to get correct temperature from the internal sensor.
  - Improvements
    * Separated lpadc_conversion_resolution_mode_t with related feature macro.
  - Bug Fixes
    * Fixed the violations of MISRA C-2012 rules:
      · Rule 10.3, 10.4, 10.6, 10.7 and 17.7.
- 2.1.1
  - Improvements
    * Updated the gain calibration formula.
    * Used feature to segregate the new item kLPADC_TriggerPriorityPreemptSubsequently.
- 2.1.0
  - New Features
    * Added the API LPADC_SetOffsetValue() to support configure offset trim value manually.
    * Added the API LPADC_DoOffsetCalibration() to do offset calibration independently.
  - Improvements
    * Improved the usage of macros and removed invalid macros.
- 2.0.2
  - Improvements
    * Added support for platforms with 2 FIFOs and different calibration measures.
- 2.0.1
  - Bug Fixes
    * Ensured the API LPADC_SetConvCommandConfig configure related registers correctly.
- 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 FIFO. 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_MasterTransferNonBlocking.

- 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_MasterTransferPrepareEDMA 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/eDMA transmission finishes.
  - New Features
    - Added APIs LPUART_GetRxFifoCount/LPUART_GetTxFifoCount to get rx/tx FIFO data count.
    - Added APIs LPUART_SetRxFifoWatermark/LPUART_SetTxFifoWatermark to set rx/tx FIFO water mark.
- 2.4.1
  - Bug Fixes
    - Fixed MISRA advisory 17.7 issues.
- 2.4.0
  - New Features
    - Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.1
  - Bug Fixes
    - Fixed MISRA advisory 15.5 issues.
- 2.3.0
  - Improvements
    - Modified LPUART_TransferHandleIRQ so that txState will be set to idle only when all data has been sent out to bus.
    - Modified LPUART_TransferGetSendCount so that this API returns the real byte count that LPUART has sent out rather than the software buffer status.
    - Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.8
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule-10.3, rule-14.4, rule-15.5.
    - Eliminated Pa082 warnings by assigning volatile variables to local variables and using local variables instead.
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 10.8, 14.4, 11.6, 17.7.
  - Improvements
    - Added check for kLPUART_TransmissionCompleteFlag in LPUART_WriteBlocking, LPUART_TransferHandleIRQ, LPUART_TransferSendDMACallback and LPUART_SendEDMACallback to ensure all the data would be sent out to bus.
    - Rounded up the calculated sbr value in LPUART_SetBaudRate and LPUART_Init to achieve more acurate baudrate setting. Changed osr from uint32_t to uint8_t since osr's bigest value is 31.
    - Modified LPUART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
- 2.2.7
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.

  - Fixed rule-12.1, rule-17.7, rule-14.4, rule-13.3, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3.

- 2.2.6
  - Bug Fixes
    - Fixed the issue of register's being in repeated reading status while dealing with the IRQ routine.
- 2.2.5
  - Bug Fixes
    - Do not set or clear the TIE/RIE bits when using LPUART_EnableTxDMA and 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

MECC

The current FLEXRAM driver version is 2.1.0.

- 2.1.0
  - Bug fixes:
    - Removed Ocram1StartAddress, Ocram1EndAddress, Ocram2StartAddress, Ocram2-EndAddress in mecc_config_t structure. Use startAddress and endAddress as instead.
    - Removed static function MECC_GetInstance().
  - New Features:
    - Added new function MECC_GetPendingFlags().
    - Added new members: enableReadDataWait, enableReadAddrPipeline, enableWrite-DataPipeline, enableWriteAddrPipeline in mecc_config_t structure to support pipeline features.
- 2.0.2
  - Bug fixes:
    - Fixed MISRA 2012 issue: 10.3, 10.4.
- 2.0.1
  - Bug fixes:
    - Fixed MISRA 2012 issue: 10.1, 10.3, 10.4, 10.6.
- 2.0.0
  - Initial version.

CSI2RX

The current CSI2RX driver version is 2.0.4.

- 2.0.4
  - Improvements
    - Updated for new format MIPI_CSI2RX_Type definition.
- 2.0.3
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules: 3.1, 10.3, 10.4, 10.8, 17.7.
- 2.0.2
  - Improvements
    - Updated to support MIMX8QX C0 header file.
- 2.0.1
  - Improvements
    - Updated to support platforms that don't have dedicated MIPI CSI2RX CSR.
  - Bug Fixes
    - Fixed the issue that the register bit PRG_RXHS_SETTLE set to wrong value.
- 2.0.0
  - Initial version.

MIPI_DSI

The current MIPI_DSI driver version is 2.2.4.

- 2.2.4
  - Bug Fixes
    - Updated the DPI setting to use float for coefficient value for more accurate calculation.
- 2.2.3
  - Bug Fixes
    - Fixed the DSI_TransferNonBlocking no interrupt issue.
    - Fixed the violations of MISRA 2012 advisory rules.
- 2.2.2
  - Bug Fixes
    - Fixed the DPI horizontal timing setting issue.
    - Fixed MISRA issue
- 2.2.1
  - Bug Fixes
    - Fixed the bug that runs to hardfault when sending long packet with 4-byte unaligned address.
- 2.2.1
  - Improvements
    - Supported long package read.
- 2.2.0
  - Improvements

        * Change parameter MIPI_DSI_Type pointer to const type.
- 2.1.0
  - Initial version.

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.

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.

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.

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.

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 PWM-X 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.

PUF

The current PUF driver version is 2.1.6.

- 2.1.6
  - Changed wait time in PUF_Init(), when initialization fails it will try PUF_Powercycle() with shorter time. If this shorter time will also fail, initialization will be tried with worst case time as before.
- 2.1.5
  - Use common SDK delay in puf_wait_usec().
- 2.1.4
  - Replace register uint32_t ticksCount with volatile uint32_t ticksCount in puf_wait_usec() to prevent optimization out delay loop.
- 2.1.3
  - Fix MISRA C-2012 issue.
- 2.1.2
  - Update: Add automatic big to little endian swap for user (pre-shared) keys destinated to secret hardware bus (PUF key index 0).

- 2.1.1
  - Fix ARMGCC build warning .
- 2.1.0
  - Align driver with PUF SRAM controller registers on LPCXpresso55s16.
  - Update initizalition logic .
- 2.0.3
  - Fix MISRA C-2012 issue.
- 2.0.2
  - New feature:
    - Add PUF configuration structure and support for PUF SRAM controller.
  - Improvements:
    - Remove magic constants.
- 2.0.1
  - Bug Fixes:
    - Fixed puf_wait_usec function optimization issue.
- 2.0.0
  - Initial version.

PXP

The current PXP driver version is 2.6.1.

- 2.6.1
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.6.0
  - Bug Fixes
    - Added missing configuration option for fetch engine background value.
    - Fixed bug in PXP_SetStoreEngineConfig that the address increment for store mask is not linear.
    - Added channel aribitration configuration for fetch engine, channel combine for store engine.
    - Fixed wrong method of obtaining the store mask address.
    - Fixed wrong method of configuring flag shift mask/width which can only be written in word boundary.
    - Fixed wrong configurations of block store and pitch in PXP_SetStoreEngineConfig.
    - Fixed wrong method of obtaining cfaValue address and calculating word count.
    - Fixed the channel word order cannot be updated when configuring the second channel.
    - Fixed bugs in PXP_SetHistogramConfig of wrong method to obtain the store mask address and wrong access of 32-bit registers.
- 2.5.0
  - New Features
    - Added new API PXP_GetPorterDuffConfigExt for flexible Porter-Duff configuration.
    - Added enumerations for new AS/PS pixel formats for certain SoCs.

- 2.4.1
  - New Features
    - Added API PXP_ResetControl to reset the PXP and the control register to initialized state.
- 2.4.0
  - New Features
    - Added the API PXP_BuildRect of building a solid rectangle of given pixel value.
    - Added the interrupt enable/disable and status mask for V3.
    - Added API PXP_EnableProcessEngine to enable/disable process engines for V3.
    - Added API PXP_SetHistogramSize to re-configure the histogram size for each update.
    - Updated PXP_WfeaInit and PXP_SetWfeaConfig according to header file's update of WFE related registers.
    - Updated PXP_WfeaInit to support handshake with upstream dither store engine and added API PXP_WfeaEnableDitherHandshake to enable/disable the feature.
    - Added API PXP_GetLutUsage to get the occupied LUT list.
    - Updated APIs to support alpha blending engine1.
    - Added the API PXP_MemCopy to support all memory size copy.
  - Bug Fixes
    - Fixed wrong naming for mux16.
    - Fixed wrong naming for enumerations in pxp_scanline_burst_t.
    - Fixed bug in PXP_GetHistogramMatchResult since there are 2 histograms engines rather than 1.
    - Fixed bug in PXP_SetFetchEngineConfig that the fetch size should not be minus one coding.
- 2.3.0
  - New Features
    - Added the configuration of fetch engine, store engine, pre-dither engine and histogram block.
- 2.2.2
  - Improvements
    - Disable alpha surface (AS) in PXP_Init.
- 2.2.1
  - Improvements
    - Added memory address conversion to support buffers which could only be accessed using alias address by non-core masters.
- 2.2.0
  - Bug Fixes
    - Fixed Porter Duff configuration error.
- 2.1.0
  - New Features
    - Added Porter Duff support.
    - Added APIs PXP_StartMemCopy and PXP_StartPictureCopy.
    - Added API PXP_SetProcessSurfaceYUVFormat.
- 2.0.2
  - Bug Fixes

        * Fixed violations of the MISRA C-2012 rules 3.1, 10.8, 11.6, 12.2.
- 2.0.1
  - Bug Fixes
    * Fixed the rotate function issue for i.MX 6ULL.
- 2.0.0
  - Initial version.

QTMR

The current QTMR driver version is 2.2.2.

- 2.2.2
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rules: 10.1, 10.8.
- 2.2.1
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rules: 10.1, 10.8.
- 2.2.0
  - Improvements
    * Added API QTMR_SetPwmOutputToIdle to set the generated pwm signal to the configured idle value.
    * Added API QTMR_GetPwmOutputStatus to return the output status of the generated pwm signal.
    * Added API QTMR_GetPwmChannelStatus to return the channel dutycycle value.
    * Added API QTMR_SetPwmClockMode to set clock mode change peripheral clock frequency.
  - Bug Fixes
    * Fixed the issue that pwm duty cycle could not be 0 and 100.
- 2.1.0
  - Bug Fixes
    * Fixed the issue QTMR_SetTimerPeriod needs to decrement down count by 1, and added new APIs to configure the LOAD register, COMP register.
- 2.0.2
  - Bug Fixes
    * Fixed the issue introduced by previous code correction for improving the output signal accuracy.
- 2.0.1
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rules: 10.1, 10.3, 11.5, 11.9.
  - Improvements
    * Improved the output signal accuracy.
- 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.

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.

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.

SEMC

The current SEMC driver version is 2.7.0.

- 2.7.0
    - Improvements
        - Add new autofreshTimes parameter in semc_sdram_config_t.
    - Bug Fixes
        - Fixed violations of MISRA C-2012 rule 10.4.
- 2.6.0
    - Bug Fixes
        - Fixed the SEMC SRAM function bug that some configuration options can’t be set.
        - Correct legacy SEMC SRAM function feature macros.
    - Improvements
        - Add new SEMC SRAM function feature macros.
- 2.5.1
    - Bug Fixes

    - Fixed violations of the MISRA C-2012 Rule 14.3.
    - Fixed SEMC_ConfigureDBI bug that RDX not set correctly.
- 2.5.0
  - Bug Fixes
    - Fixed definitions of bitfields of BMCR0 and BMCR1 - wrong field order and incorrect semantical naming
    - The fix alters the driver API regarding configuration of AXI bus queue reordering
- 2.4.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 5.6.
- 2.4.2
  - Improvements
    - Deleted meaningless parameter in memory size conversion function.
- 2.4.1
  - Bug Fixes
    - Fixed PSRAM A8 configuration issue, which should be 0x06U for PSRAM while pix mux bit width is 0x04U, based on different pix mux bit width.
- 2.4.0
  - Improvements
    - Improved nor and sram timing configuration on sync mode.
- 2.3.1
  - Bug Fixes
    - Updated refresh timer period(RT) timing setting, which updated into (RT+1)*(Prescaler period) for SDRAM.
    - Supported new DBI control register 2 to configure CSX interval time(CEITV).
    - Fixed violations of the MISRA C-2012 Rule 10.8.
    - Fixed doxygen warning.
- 2.3.0
  - New Features
    - Limited burst length as 1 according to ERR050577, Auto-refresh command may possibly fail to be triggered during long time back-to-back write (or read) when SDRAM controller's burst length is greater than 1.
    - Supported 8 bits column address for SDRAM.
- 2.2.1
  - New Features
    - Added queue weight control, which can control queue a/b is working or not.
    - Updated NAND FLASH configuration API which disables and enables SEMC between configure control registers.
    - Added ONFI parameter Integrity CRC check for SEMC flash component.
- 2.2.0
  - New Features
    - Supported up to 4 PSRAM CS.
    - Added programmable delay line for DQS.
    - Added ready/wait feature for SRAM in asynchronous mode.
- 2.1.0

  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3, 10.4, and 14.4.
    - Updated parameter type from uint16_t into uint32_t for send IP command API.
- 2.0.4
  - Bug Fixes
    - Fixed the SEMC queueA and queueB weight configuration issue.
    - Fixed the wrong configuration of DBICR1 register in SEMC_ConfigureDBI.
- 2.0.3
  - Bug Fixes
    - Added feature macro to control WDS&WDH bit setting for NOR synchronous transfer.
- 2.0.2
  - Bug Fixes
    - Changed SEMC NAND configuration structure and verify SEMC NAND related APIs.
    - Added extended SEMC clock enable.
- 2.0.1
  - Bug Fixes
    - Fixed data size mask configure in SEMC_ConfigureIPCommand API.
    - Updated the command mode in IP command type.
- 2.0.0
  - Initial version.

SMARTCARD

The current SMARTCARD driver version is 2.3.0.

- 2.3.0
  - New features:
    - Added support for USIM
- 2.2.2
  - Bug fix:
    - Fixed MISRA C-2012 rule 10.4.
- 2.2.1
  - Bug fix:
    - Fixed IAR warnings Pa082 in smartcard_emvsim
    - Fixed MISRA issues
    - Fixed rules 10.1, 10.3, 10.4, 10.6, 10.7, 10.8, 14.4, 16.1, 16.3, 16.4, 17.7
- 2.2.0
  - New features:
    - Updated to use RX/TX FIFO
- 2.1.2
  - Provided time delay function which works in microseconds.
  - Bug fix:
    - Changed event to semaphore in RTOS driver (KPSDK-11634).
    - Added check if de-initialized variables are not null iSMARTCARD_RTOS_Deinit()

(KPSDK-8788).
    - Changed deactivation sequence iSMARTCARD_PHY_TDA8035_Deactivate() to properly stop the clockPOSCR-35).
    - Fixed timing issue with VSEL0/1 signals in smartcard TDA803driver (KPSDK-10160)
- 2.1.1
  - New features:
    - Added default phy interface selection into smartcard RTOS drivers (KPSDK-9063).
    - Replaced smartcard_phy_ncn8025 driver by smartcard_phy_tda8035.
  - Bug fix:
    - Fixed protocol timers activation sequences in smartcard_emvsim and smartcard_phy_-tda8035 drivers during emvl1 pre-certification tests (KPSDK-9170, KPSDK-9556).
- 2.1.0
  - 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 NU-LL.
- 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.

SSARC

The current SSARC driver version is 2.1.0.

- 2.1.0
  - Improvements
    - Updated the structure ssarc_descriptor_config_t, make it more friendly to users.
- 2.0.0
  - Initial version.

TEMPSENSOR

The current TEMPSENSOR driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    - Fixed the bug of incorrect default value of temperature sensor registers in initialization state.
- 2.1.1
  - Improvements
    - CTRL0 register fields are not needed for customer, they are trim registers for the IP that are determined during calibration.
- 2.1.0
  - Improvements
    - Supported directly access to TEMPSENSOR registers.
- 2.0.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.8.
- 2.0.2
  - Bug Fixes
    - Fixed bug that FINISH flag not cleared after temperature read out.
- 2.0.1
  - Improvements
    - Updated temperature calculation formula, to get more accurate result with high or low temperature..
- 2.0.0
  - Initial version.

USDHC

The current USDHC driver version is 2.8.4.

- 2.8.4
  - Improvements
    - Add feature macro FSL_FEATURE_USDHC_HAS_NO_VS18.
- 2.8.3
  - Improvements
    - Improved api USDHC_EnableAutoTuningForCmdAndData to adapt to new bit field name for USDHC_VEND_SPEC2 register.
- 2.8.2
  - Improvements
    - Added feature macro FSL_FEATURE_USDHC_HAS_NO_VOLTAGE_SELECT.
- 2.8.1
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9.
- 2.8.0
  - Improvements
    - Fixed the mmc boot transfer failed issue which is caused by the Dma complete interrupt not enabled.
    - Marked api USDHC_AdjustDelayForManualTuning as deprecated and added new api USDHC_SetTuingDelay/USDHC_GetTuningDelayStatus.
    - Improved the manual tuning flow accroding to specification.
    - Added memory address conversion to support buffers which could only be accessed using alias address by non-core masters.
    - Fixed violations of MISRA C-2012 rule 10.4.
- 2.7.0
  - Improvements
    - Added api USDHC_TransferScatterGatherADMANonBlocking to support scatter gather transfer.
    - Added feature FSL_FEATURE_USDHC_REGISTER_HOST_CTRL_CAP_HAS_NO_RETUNING_TIME_COUNTER for re-tuning time counter field in HOST_CTRL_CAP register.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9, 10.1, 10.3, 10.4, 8.4.
- 2.6.0
  - Improvements
    - Added api USDHC_SetStandardTuningCounter to support adjust tuning counter of Standard tuning.
- 2.5.1
  - Improvements
    - Used different status code for command and data interrupt callback.
    - Added cache line invalidate for receive buffer in driver IRQ handler to fix CM7 speculative access issue.
- 2.5.0
  - Improvements
    - Added new api USDHC_SetStrobeDllOverride for HS400 strobe dll override mode delay taps configurations.

    - Corrected the STROBE DLL configurations sequence.
- 2.4.0
  - Improvements
    - Added feature macro for read/write burst length.
      - Disabled redundant interrupt per different transfer request.
      - Disabled interrupt and reset command/data pointer in handle when transfer completes.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
    - Fixed PA082 build warning.
    - Fixed logically dead code Coverity issue.
- 2.3.0
  - Improvements
    - Added USDHC_SetDataConfig API to support manual tuning.
    - Removed the limitaion that source clock must be bigger than the target in function USDHC_SetSdClock by using source clock frequency as target directly.
    - Added peripheral reset in USDHC_Init function.
    - Added tuning reset support in function USDHC_Reset function.
- 2.2.8
  - Bug Fixes
    - Fixed out-of bounds write in function USDHC_ReceiveCommandResponse.
- 2.2.7
  - Improvements
    - Added API USDHC_GetEnabledInterruptStatusFlags and used in USDHC_TransferHandleIRQ.
    - Removed useless member interruptFlags in usdhc_handle_t.
- 2.2.6
  - Improvements
    - Added address align check for ADMA descriptor table address.
    - Changed USDHC_ADMA1_DESCRIPTOR_MAX_LENGTH_PER_ENTRY to (65536-4096) to make sure the data address is 4KB align for a transfer which need more than one ADMA1 descriptor.
- 2.2.5
  - Bug Fixes
    - Fixed MDK 66-D warning.
- 2.2.4
  - Bug Fixes
    - Fixed issue that real clock frequency wss mismatched with target clock frequency, which was caused by an incorrect prescaler calculation.
  - New Features
    - Added control macro to enable/disable the CLOCK code in current driver.
- 2.2.3
  - Bug Fixes
    - Fixed issue where AMDA did not disable with DMAEN clear.

  - Improvements
    - Improved set clock function to check the output frequency range.
    - Dynamic set SDCLKFS during DDR enable or disable.
- 2.2.2
  - Improvements
    - Improved read transfer cache maintain operation, combined clean, and invalidated them into one function.
- 2.2.1
  - Bug Fixes
    - Disabled the invalidate cache operation for tuning.
- 2.2.0
  - Improvements
    - Improved USDHC to support MMC boot feature.
- 2.1.3
  - Bug Fixes
    - Fixed MISRA issue.
- 2.1.2
  - Bug Fixes
    - Fixed Coverity issue.
    - Added base address and userData parameter for all callback functions.
- 2.1.1
  - Improvements
    - Added cache maintain operation.
    - Added timeout status check for the DATA transfer which ignore error.
    - Added feature macro for SDR50/SDR104 mode.
    - Removed useless IRQ handler from different platforms.
- 2.1.0
  - Improvements
    - Integrated tuning into transfer function.
    - Added strobe DLL feature.
    - Added enableAutoCommand23 in data structure.
    - Removed enable card clock function because the controller would handle the clock on/off.
- 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.

XBARB

The current XBARB driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 12.2, 10.7
- 2.0.1
  - Bug Fixes
    * Corrected XBARB_SetSignalsConnection function.
  - Other Changes
    * Changed array clock name.
- 2.0.0
  - Initial version.

XECC

The current XECC driver version is 2.0.0.

- 2.0.0
  - Initial version.

XRDC2

The current XRDC2 driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    * Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.8, 12.2, 14.1.
- 2.0.1
  - Improvements
    * Updated for new header file.
- 2.0.0
  - Initial version.

2 CANopen Change Log

CANopen for KSDK

- 7.10_rev1
  - update version
- 7.01_rev1
  - update version
- 7.00_rev1
  - Integrate CANopen 7.00 to SDK.

flash management stack for KSDK

Current driver version is 2.0.0

- 2.0.0 Initial version, remove the bb_last initial value.

eIQ TensorFlow Lite for Microcontrollers library

Current version is 23-09-18

- 22-09-18
  - New Features
    - Updated eIQ TensorFlow Lite for Microcontrollers library to version from the 18th of September 2023
    - Updated third party library source codes:
      - CMSIS-NN
- 22-03-23
  - New Features
    - Updated eIQ TensorFlow Lite for Microcontrollers library to version from the 23rd of March 2023
    - Updated third party library source codes:
      - CMSIS-NN
      - Xtensa NN library
- 22-09-19
  - New Features
    - Updated eIQ TensorFlow Lite for Microcontrollers library to version from the 19th of September 2022
    - Updated third party library source codes:
      - CMSIS-NN
      - FFT2D
      - FlatBuffers
      - Gemmlowp

        - Ruy
        - Xtensa NN library
    - Improvements
        - Shortened example names
- 22-02-16
    - New Features
        - Updated eIQ TensorFlow Lite for Microcontrollers library to version from the 16th of February 2022
        - Updated third party library source codes:
            - CMSIS-NN
            - FFT2D
            - FlatBuffers
            - Gemmlowp
            - Ruy
            - Xtensa NN library
- 2.6.0
    - New Features
        - Updated eIQ TensorFlow Lite for Microcontrollers library to version 2.6.0
        - Updated third party library source codes:
            - CMSIS-NN
            - FFT2D
            - FlatBuffers
            - Gemmlowp
            - Ruy
            - Xtensa NN library
- 2.4.1
    - New Features
        - Updated eIQ TensorFlow Lite for Microcontrollers library to version 2.4.1
        - Removed source codes related to TensorFlow Lite library only
        - Updated third party library source codes:
            - CMSIS-NN
            - FFT2D
            - FlatBuffers
            - Gemmlowp
            - Ruy
        - Converted TensorFlow Lite examples to TensorFlow Lite for Microcontrollers (changed API use)
- 2.3.1
    - New Features
        - Updated eIQ TensorFlow Lite library to version based on the TensorFlow Lite library version 2.3.1
        - Added TensorFlow Lite for Microcontrollers library source codes
        - Updated third party library source codes:
            - Abseil
            - Eigen

        - Farmhash
        - FFT2D
        - FlatBuffers
        - Gemmlowp
      - Added third party library source codes:
        - CMSIS-NN
        - Ruy
      - Added examples:
        - tensorflow_lite_micro_label_image
- 2.1.0
  - New Features
    - Updated eIQ TensorFlow Lite library to version based on the TensorFlow Lite library version 2.1.0
    - Updated third party library source codes:
      - Eigen
      - Farmhash
      - FFT2D
      - FlatBuffers
      - Gemmlowp
    - Added third party library source codes:
      - Abseil
    - Added examples:
      - tensorflow_lite_benchmark
- 1.14.0
  - New Features
    - Updated eIQ TensorFlow Lite library to version based on the TensorFlow Lite library version 1.14.0
    - Updated third party library source codes:
      - Eigen
      - Farmhash
      - FFT2D
      - FlatBuffers
      - Gemmlowp
    - Added examples:
      - tensorflow_lite_adt
  - Improvements
    - Extended examples:
      - tensorflow_lite_cifar10
      - Added camera and LCD support
      - Realtime camera image inference
      - tensorflow_lite_kws
      - Added microphone and headphone support
      - Realtime audio inference
      - tensorflow_lite_label_image
      - Added camera and LCD support

        - Realtime camera image inference
- 1.11.0
  - New Features
    - Added eIQ TensorFlow Lite library based on TensorFlow Lite version 1.11.0
    - Added third party library source codes:
      - Eigen
      - Farmhash
      - FFT2D
      - FlatBuffers
      - Gemmlowp
    - Added examples:
      - tensorflow_lite_cifar10
      - tensorflow_lite_kws
      - tensorflow_lite_label_image
      - tensorflow_lite_lib

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

LigJpeg for KSDK

Current version is LigJpeg 9b.

- 9b_rev1
  - New Feature:
    - The configuration file libjpeg/inc/jmorecfg.h could include user defined header file to override pixel format configuration.
- 9b_rev0
  - Initial version. Changes when integrate with SDK:
    - In libjpeg/inc/jinclude.h line 88-96, map JFREAD and JFWRITE to FATFS f_read and f_write
    - In libjpeg/inc/jmorecfg.h line 397-406, change RGB color offset.
    - In libjpeg/src/jerror.c line 79-81, don't call function exit.

fail-safe filesystem for MCUXpresso SDK

The current version littlefs filesystem is 2.5.0_rev0.

- 2.9.1_rev0
  - littlefs updated to version 2.9.1
- 2.8.0_rev0
  - littlefs updated to version 2.8.0
- 2.5.0_rev0
  - littlefs updated to version 2.5.0
- 2.4.1_rev0
  - littlefs updated to version 2.4.1
- 2.4.0_rev0
  - littlefs updated to version 2.4.0
- 2.2.1_rev0
  - littlefs updated to version 2.2.1
- 2.1.4_rev0
  - littlefs updated to version 2.1.4
- 1.3_rev0
  - Initial version of littlefs filesystem for MCUXpresso SDK

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.

lwIP for MCUXpresso SDK

Lightweight IP (lwIP) is a small independent implementation of the TCP/IP protocol suite. Source code included in this SDK is based on development version 2.2.1 taken from 3rd party lwIP GIT repository. The webpage https://git.savannah.nongnu.org/cgit/lwip.git allows to browse the repository and also contains URLs for its cloning. The development versions (X.Y.Z.dev) do not refer to a single source code snapshots. To avoid ambiguity, change log below contains SHA-1 hashes of GIT commits used when importing the code into the SDK.

- 2.2.1_rev1
  - New features:
    - Ported lwIP 2.2.1.dev (2024-02-19, branch: master, SHA-1: d0efd9ef7ba08e54b46b1060e2b4629a
    to MCUXpresso SDK.
    - Added ETH_MAX_RX_PKTS_AT_ONCE macro. See port/README.md for details.
    - In port/netc_ethernetif.c, added NETC_VSI_NUM_USED macro to support using VSI. A thread of SI message handling will be started to handle VSI-PSI messages.
  - Bug fixes:
    - Added the missing implementation for IP_FORWARD_ALLOW_TX_ON_RX_NETIF option in the function ip6_forward. Therefore IPv6 packets could be sent back out on the netif where they where originally received from.
    - NETC adaptation layer: Do not call xEventGroupSetBits from ISR.
  - Ethernet adaptation layers: Default value of priority of the receive task (ETH_RX_TASK_PRIO) is set lower than the priority of the FreeRTOS daemon task (timer task).
- 2.2.0_rev11
  - New features:
    - NETC adaptation layer: Possible to disable IPv4/TCP/UDP checksum validation done

in HW.
    - EtherCAT EoE(Ethernet over EtherCAT) driver is added to lwip.
  - Bug fixes:
    - src/apps/httpsrv/httpsrv_supp.c: Fixed performing of the HTTP server task priority limitation.
- 2.2.0_rev10
  - New features:
    - Ported lwIP 2.2.0 (2023-09-25, branch: master, SHA-1: 0a0452b2c39bdd91e252aef045c115f88f6c tag: STABLE-2_2_0_RELEASE) to MCUXpresso SDK.
    - Enabled hardware-accelerated CRC computation and verification (MAC, IPv4, TCP, UDP, ICMPv4, ICMPv6) for ENET Kinetis, ENET QoS and ENET LPC.
    - Enabled link state detection based on PHY interrupts. The ETH_LINK_POLLING_INTERVAL_MS macro controls this - setting it to 0 and specifying ethernetif_config_t->phyIntGpio enables it, setting it to a value greater than zero enables polling instead. Supported only under an RTOS (NO_SYS == 0). By default, the link state is polled.
    - ND6: Implemented RFC 4191 type C host, which means default router list (learned from Router Advertisement messages) has been replaced with routing table, which contains default route records for each router and also routes learned from received Route Information Options. Changes partially based on https://savannah.nongnu.org/patch/?10114. The option LWIP_ND6_NUM_ROUTERS has been removed, and the new option LWIP_ND6_NUM_ROUTES has been added to configure the size of the routing table.
    - IPv6: Implemented a new hook - LWIP_HOOK_IP6_CANFORWARD. This hook can be used, for example, for multicast forwarding between netifs. Defining this hook enables multicast traffic forwarding, thus the hook is also invoked for multicast traffic.
    - MLD6: Multicast Listener Discovery v1 replaced by v2 (RFC 3810) but without support of source specific multicast.
    - port/enet_ethernetif_kinetis.c: Added check to generate/validate ICMPv6 checksum in SW as the Kinetis ENET peripheral does not do it.
    - Added disabling of Rx interrupt when the port is out of Rx buffers. See port/README.md for more details.
  - Bug fixes:
    - src/apps/lwiperf: Fixed access to invalid data when UDP report is to be sent from a timer but abort has been called before.
    - src/apps/lwiperf: Fixed deallocation of TCP server started by client (in reverse or dual modes) which failed to connect.
    - port/netc_ethernetif.c: Fixed cache control enablement macro (FSL_SDK_ENABLE_DRIVER_CACHE_CONTROL > FSL_ETH_ENABLE_CACHE_CONTROL).
    - port/sys_arch.c: The function sys_assert does not call portENTER_CRITICAL when called from an interrupt.
    - src/core/ipv4/ip4.c: Fixed checksum reset condition.
    - ND6:
      - Lladdr length is now taken from netif->hwaddr_len so ND6 works properly regardless of NETIF_MAX_HWADDR_LEN.
      - Added check of sufficient length of lladdr options from incoming messages.

    - src/apps/httpsrv/httpsrv.c: Fixed hangup in HTTPSRV_release if caller's task has higher priority than server task.
  - port/arch/cc.h: LWIP_PLATFORM_DIAG is defined (and can be overridden) independently of the LWIP_DEBUG setting. Removed printing extra newline symbols from LWIP_PLATFORM_DIAG.
  - src/apps/lwiperf: The "end of test" UDP datagram is resent more often. This increases the probability of the server to receive it and end the test when datagrams are getting lost.
  - Added port/README.md describing possible settings and helper functions in the port layer.
- 2.2.0_rev9
  - New features:
    - Ported lwIP 2.2.0.dev (2023-01-03, branch: master, SHA-1: 3fe8d2fc43a9b69f7ed28c63d44a7744f to MCUXpresso SDK.
    - Applied patch to allow sending IPv6 router advertisement. Improved to allow selection of interface and router life time and to allow sending route information options.
    - src/apps/lwiperf: Support for reverse test (client receives, server sends). Requires iperf version 2.1.0 or newer.
  - Bug fixes:
    - src/apps/httpsrv: Fixed operation with LWIP_IPV6 enabled. Server can be also accessed using both IPv4 and IPv6 at the same time if compiled with both LWIP_IPV4=1 and LWIP_IPV6=1. Note the type of the field struct httpsrv_param_struct.address has changed from struct sockaddr to struct sockaddr_storage.
- 2.2.0_rev8
  - New features:
    - src/apps/lwiperf: Added new parameter "buffer_len" to functions lwiperf_start_tcp_client() and lwiperf_start_udp_client() to configure TCP/UDP packet size.
    - src/apps/lwiperf: Added new parameter "tos" to functions lwiperf_start_tcp_client() to configure TCP packet priority.
    - NETC adaptation layer: Not forcing the RX/TX buffers placement in non-cacheable memory. Requires the symbol FSL_ETH_ENABLE_CACHE_CONTROL to be defined on project level if the memory region, where the buffers are placed by a linker, has cache enabled.
  - Bug fixes:
    - src/apps/httpsrv: Added missing includes.
    - src/apps/lwiperf: Fixed TCP client to send settings at the beginning of each 128 KB block like the PC iperf 2.0.x application does.
    - src/apps/lwiperf: Fixed validation of TCP received data (with LWIPERF_CHECK_RX_DATA enabled, works with iperf 2.0.x).
    - src/apps/lwiperf: Fixed lwiperf_list_remove() to clear references to the removed item.
  - src/apps/lwiperf: Program does not assert when buffer cannot be cloned in UDP test, only "can't clone buffer" message is printed.
- 2.2.0_rev7
  - New features:
    - Ported lwIP 2.2.0.dev (2022-05-09, branch: master, SHA-1: 239918ccc173cb2c2a62f41a40fd893f5 to MCUXpresso SDK.
    - Added function ethernetif_probe_link() which reads actual link, speed and duplex

settings from phy and passes them to driver. Stack could be set to call this function periodically by setting ETH_LINK_POLLING_INTERVAL_MS to value higher than zero.
    - Added helper functions ethernetif_wait_linkup() and ethernetif_wait_ipv4_valid() to allow blocking of RTOS task or bare metal application until link is up or IPv4 address becomes valid.
    - Added NETC adaptation layer.
    - Processing of rx packets under RTOS moved from ISR to a separate task to improve system reaction times. Switch back to old behavior can be done by setting ETH_DO_-RX_IN_SEPARATE_TASK macro to 0.
  - Bug fixes:
    - port: Fixed copying of pbuf contents. Previous code was using an incorrect end condition and could result in the overrun of the destination buffer if more packets were on the queue.
    - port: Delegating pbuf_free calls to tcpip_thread via pbuf_free_callback where possible (RTOS), ensured pbuf_free is not called from interrupt context when LWIP_ALLOW-_MEM_FREE_FROM_OTHER_CONTEXT is not set (bare metal).
    - port/enet_ethernetif_qos.c - Fixed ENET_RXBD_NUM which was used instead of E-NET_TXBD_NUM.
    - port/enet_ethernetif_qos.c - Fixed buffer alignment to be at least 64.
    - src/apps/lwiperf: Fixed IPv6 TCP TX throughput lower than IPv4 by modifying maximum segment size to avoid sending two segments instead of one.
    - src/apps/lwiperf: Out-of-order datagrams in UDP RX server mode are counted to the throughput.
    - src/apps/httpsrv: Implemented receive timeouts on sockets.
    - src/apps/httpsrv: Don't assert on HTTP session task creation failure.
    - src/apps/httpsrv: Fixed build with IPv6 enabled.
    - src/apps/httpsrv: Updated endianess macros required for websocket SHA generation.
    - src/apps/httpsrv: Added missing includes.
- 2.2.0_rev6
  - New features:
    - Ported lwIP 2.2.0.dev (2022-03-25, branch: master, SHA-1: 124dc0a64ef5d7c14a27e3115e5888df
    to MCUXpresso SDK.
    - Implemented leaving of multicast groups on ENET and ENET QOS.
- 2.2.0_rev5
  - New features:
    - Ported lwIP 2.2.0.dev (2021-05-11, branch: master, SHA-1: 7ec4e9be304e7f8953740f10b2c810a2
    to MCUXpresso SDK.
    - LPC ENET adaptation layer allocates more buffers for frame reception now. Previously the number of receive buffers was determined by ENET_RXBD_NUM, which defaults to 5. It is determined by ENET_RXBUFF_NUM now, which is 2 ∗ ENET_RXBD_N-UM by default. Increase was needed because the actual version of LPC ENET driver always hold ENET_RXBD_NUM number of buffers and few additional buffers are needed for passing zero-copy frame data to lwIP. If this takes too much memory in your application, you can counteract by decreasing PBUF_POOL_SIZE, since PBUF_-

POOL is used only for transmission when LPC ENET, Kinetis ENET or ENET QOS is used.

- 2.2.0_rev4
  - New features:
    - Ported lwIP 2.2.0.dev (2021-03-05, branch: master, SHA-1: 0056522cc974d2be2005c324f37187b5 to KSDK 2.0.0.
    - LWIP_DHCP_DOES_ACD_CHECK option default changed to 0 (disabled):
      - Although the ACD check makes getting IP address from DHCP more robust, it added several seconds delay at startup of all applications which use DHCP.
      - This feature was not present in earlier versions of lwIP.
    - ENET QOS adaptation layer - implemented zero-copy on receive.
    - Kinetis ENET and ENET QOS adaptation layers allocate more buffers for frame reception now. Previously the number of receive buffers was determined by ENET_RXBD_NUM, which defaults to 5. It is determined by ENET_RXBUFF_NUM now, which is 2 ∗ ENET_RXBD_NUM by default. Increase was needed because the actual version of Kinetis ENET and ENET QOS drivers always hold ENET_RXBD_NUM number of buffers and few additional buffers are needed for passing zero-copy frame data to lwIP. If this takes too much memory in your application, you can counteract by decreasing PBUF_POOL_SIZE, since PBUF_POOL is used only for transmission when Kinetis ENET or ENET QOS is used.
    - Removed ethernetif_config_t.non_dma_memory field which was required to configure memory ranges unusable by ENET DMA on LPC devices. The setting has been replaced by BOARD_ENET_NON_DMA_MEMORY_ARRAY macro.
- 2.2.0_rev3
  - New features:
    - Ported lwIP 2.2.0.dev (2020-07-07, branch: master, SHA-1: c385f31076b27efb8ee37f00cb556878 to KSDK 2.0.0.
- 2.2.0_rev2
  - New features:
    - Kinetis ENET adaptation layer - implemented zero-copy on receive.
    - lwiperf - counter of transferred bytes extended from 32 to 64 bit
  - Bug fixes:
    - Fixed restarting Auto IP from DHCP.
- 2.2.0_rev1
  - New features:
    - Ported lwIP 2.2.0.dev (2019-12-12, branch: master, SHA-1: 555812dcec38c9a2ef1ef9b318162915 to KSDK 2.0.0.
    - Implemented LWIP_ASSERT_CORE_LOCKED related functions in sys_arch.c. It can be enabled in lwipopts.h:
      - `#define LWIP_ASSERT_CORE_LOCKED() sys_check_core_locking()`
      - `#define LWIP_MARK_TCPIP_THREAD() sys_mark_tcpip_thread() // if NO_SYS == 0`
      - `#define LOCK_TCPIP_CORE() sys_lock_tcpip_core() // if NO_SYS == 0 and LWIP_TCPIP_CORE_LOCKING == 1`

```
· #define UNLOCK_TCPIP_CORE() sys_unlock_tcpip_core()
  // if NO_SYS == 0 and LWIP_TCPIP_CORE_LOCKING == 1
```

- 2.1.2_rev5
  - New features:
    - Implemented TCP_USER_TIMEOUT socket option.
    - Implemented SIOCOUTQ ioctl.
- 2.1.2_rev4
  - New features:
    - Ported lwIP 2.1.3.dev (2019-02-27, branch: STABLE-2_1_x, SHA-1: 1bb6e7f52de1cd86be0eed31 to KSDK 2.0.0.
    - Updated sys_thread_new implementation and comment.
    - Kinetis ENET adaptation layer - reading frames into a pbuf chain is conditionally compiled only when a single pbuf from pool cannot hold maximum frame size (PBUF_POOL_BUFSIZE >= maximum frame size). Avoiding this code also reduces stack size requirements by about 1.5 kilobytes.
  - Bug fixes:
    - Fixes in ethernetif_linkoutput() in enet_ethernetif_lpc.c:
      - Removed access to possibly freed pbuf.
      - Call pbuf_free() when transmit buffers not available.
      - When copying pbuf chain, updating the number of necessary transmit buffers to wait for, which can be often smaller in the copy.
    - When CGI script is reading POST data by chunks, the loop in httpsrv_read() may cause blocking in receive function waiting for more data at the end of the stream
      - HTTPSRV_cgi_read() - added limiting of the last chunk length according to content length to avoid undesired blocking
    - Applied AUTOIP patch https://savannah.nongnu.org/patch/?9847 - with modification to support multiple network interfaces.
    - Fixed buffer overflow in httpsrv when application provided CGI script does not handle the whole content of POST request
  - Removed LwipMibCompiler contrib application as it contained LGPL licensed files in SharpSnmpLib.
- 2.1.2_rev3
  - New features:
    - lwiperf updated with UDP client/server support from the patch 9751 (https://savannah.nongnu.org/patch/?9751)
- 2.1.2_rev2
  - Bug fixes:
    - Fixed lwiperf_abort() in lwiperf.c to correctly close connections and free resources
- 2.1.2_rev1
  - New features:
    - Ported lwIP 2.1.2 (2018-11-22, SHA-1: 159e31b689577dbf69cf0683bbaffbd71fa5ee10) to KSDK 2.0.0.
    - Ported lwIP-contrib 2.1.0 (2018-09-24, SHA-1: 35b011d4cf4c4b480f8859c456587a884ec9d287) to KSDK 2.0.0.
- 2.0.3_rev1

  - New features:
    - Ported lwIP 2.0.3 (2017-09-15, SHA-1: 92f23d6ca0971a32f2085b9480e738d34174417b) to KSDK 2.0.0.
- 2.0.2_rev1
  - New features:
    - Ported lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef) to KSDK 2.0.0.
- 2.0.0_rev3
  - New features:
    - Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0.
- 2.0.0_rev2
  - New features:
    - Ported lwIP 2.0.0 RC2 (2016-08-08, SHA-1: b1dfd00f9233d124514a36a8c8606990016f2ad4) to KSDK 2.0.0.
- 2.0.0_rev1
  - New features:
    - Ported lwIP 2.0.0 RC0 (2016-05-26) to KSDK 2.0.0.
    - Changed lwIP bare-metal examples to use poll-driven approach instead of interrupt-driven one.
- 1.4.1_rev2
  - New features:
    - Enabled critical sections in lwIP.
  - Bug fixes:
    - Fixed default lwIP packet-buffer size to be able to accept a maximum size frame from the ENET driver.
    - Fixed possible drop of multi-frame packets during transmission.
- 1.4.1_rev1
  - New features:
    - Ported lwIP 1.4.1 to KSDK 2.0.0.

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.

MMCAU library

The current version is 2.0.4.

- 2.0.4
  - Bug fixes:
    * Fixed issue while multiple SHA blocks passed in HASH API.
- 2.0.3
  - Bug fixes:
    * Fixed MISRA issues in fsl_mmcau.c .
- 2.0.2
  - Bug fixes:
    * Re-writing ASM MMCAU lib to be interrupt safe on Cortex-M0+ core by expanding LDM/STM instructions by a sequence of LDR and STR instructions.
- 2.0.1
  - Bug fixes:
    * KPSDK-17133 fix bug in fsl_mmcau.c when AES key schedule array is not aligned.

- 2.0.0
  - New features:
    - Q4/2013 release of the CAU library.
    - Added fsl_mmcau.h/fsl_mmcau.c optional layer between application and legacy CAU library (cau_api.h). This API has no alignment requirements.

MOTOR_CONTROL for KSDK

Current version is 1.1.0

- 1.1.0
  - Initial version.

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

OpenH264 for KSDK

Current version is OpenH264 v2.1.1.

- 2.1.1_rev0
  - Initial version. Changes when integrate with SDK are marked using the macro **NXP_MSDK**.

RTCESL for KSDK

Current version is 4.3

- 4.3
  - Initial version.

SAFETY_IEC60730B for KSDK

Current version is 1.1.0

- 1.1.0
  - Initial version.

Host USDHC driver for MCUXpresso SDK

The current driver version is 2.6.3.

- 2.6.3
  - Improvements
    - Added macro SDMMCHOST_SUPPORT_VOLTAGE_CONTROL.
- 2.6.2
  - Bug Fixes
    - Added clock force on during standard tuning to fix the card access not stable after initialization.
- 2.6.1
  - Improvements
    - Increased the delay after enable DAT3 detect card feature to fix the misdetect issue.
- 2.6.0
  - Improvements
    - Removed deprecated api in SDHC host driver.
    - Added SDMMCHOST_ConvertDataToLittleEndian api.
    - Added capability/maxBlockCount/maxBlockSize in host decriptior.
    - Improved the manual tuning flow according to specification.
    - Added mutual exclusive access for function init/deinit/reset/transfer function.
    - Fixed violations of MISRA C-2012 rule 10.1, 10.4, 16.3, 4.7.
- 2.5.3
  - Bug Fixes
    - Corrected the DAT3 detect card flow by PULL down the DAT3 pin firstly and then enable the host DAT3 function.
- 2.5.2
  - Improvements
    - Improved DAT3 card detect mechanism to avoid card false detection.
- 2.5.1
  - Improvements
    - Enabled DAT3 card detect interrupt in function SDMMCHOST_PollingCardDetectStatus to support DAT3 re-detect card.
- 2.5.0
  - Improvements
    - Added cache line size alignment maintain for the read transfer.
    - Added FSL_FEATURE_HAS_L1CACHE to enable cache maintain operation for the soc has LMEM cache.

  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.4.0
  - Improvements
    - Added cache maintain functionality in the host driver.
    - Enabled DAT3 card detect feature.
    - Increase the default STD tuning counter to 60 to cover range of the tuning window.
    - Added host instance capability macro.
    - Added clear card inserted/removed event when card removed/inserted interrupt generated.
- 2.3.0
  - Improvements
    - Merged the host controller driver from polling/freertos/interrupt to non_blocking/blocking.
    - Added SDMMC OSA layer to support muxtex access/event/delay.
- 2.2.14
  - Bug Fixes
    - Fixed uninitialized value Coverity issue.
- 2.0.0
  - Initial version

MMC Card driver for MCUXpresso SDK

The current driver version is 2.5.1.

- 2.5.1
  - Bug Fixes
    - Fix the decoding of CID register based on JEDEC Standard 84-B51.
- 2.5.0
  - Improvements
    - Added api MMC_SetSleepAwake to support enter/exit sleep state.
    - Added new api MMC_PollingCardStatusBusy for application polling card status.
    - Removed deprecated api in mmc driver and mark MMC_HostReset as deprecated.
    - Improved the read/write/erase function flow.
    - Added mutual exclusive access for init/deinit/read/write/erase function.
    - Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.7, 10.4, 13.5, 14.4, 10.6.
- 2.4.1
  - Improvements
    - Improved the voltage window argument of CMD1 according to host capabilty instead of use card ocr directly.
    - Added host HS200/HS400/8bit bus width capability validation during card initialization.
    - Used cache line size align buffer for MMC relate api.
    - Increased the CMD13 timeout count to avoid polling CMD13 time out issue.
  - Bug Fixes

    - Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.4.0
  - Improvements
    - Added new apis MMC_EnableCacheControl/MMC_FlushCache to support cache feature.
- 2.3.1
  - Improvements
    - Removed the dead loop while polling DAT0 and CMD13 instead of using timeout mechanism.
    - Added card state check before switching to HS400 to improve the emmc initialization stability.
    - Removed the redundant operation of memset internal buffer in MMC_WrtiteBlocks function.
  - Bug Fixes
    - Fixed the sandisk emmc always busy while sending CMD1 without supported voltage provide in argument.
- 2.3.0
  - Improvements
    - Deprecated api MMC_PowerOnCard/MMC_PowerOffCard by api MMC_SetCardPower.
    - Added internalBuffer in mmc_card_t and removed rawCid/rawCsd/rawExtendedCsd.
    - Added retuning support during data transfer under HS200 mode.
    - Increased the read/write blocks failed retry times for stability.
    - Added delay while retry the CMD1 for stability.
    - Added legacy card support, the card not support CMD6, CMD8.
- 2.2.13
  - Improvements
    - Used the boot mode value instead of boot mode mask value as the parameter of MMC_SetBootConfig to improve user experience.
    - Removed dynamic voltage switch feature for mmc, according to JEDEC standard, the voltage should be fixed after power up.
- 2.2.12
  - Improvement
    - Increased the CMD1 retry times in the MMC card driver to improve driver compatibility.
  - Bug Fixes
    - Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).
    - Fixed the fall through build warning by adding SUPPRESS_FALL_THROUGH_WARNING() in mmc driver.
- 2.2.7
  - Bug Fixes
    - Fixed MDK 66-D warning.

- 2.2.6
  - Improvements
    - Saved MMC OCR registers while sending CMD1 with argument 0.
  - Bug Fixes
    - Added MMC_PowerOn function in which there is delay function after powerup sdcard. Otherwise, the card initialization by fail.
- 2.2.5
  - Improvements
    - Added SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning and it is disabled by default.
- 2.2.4
  - Bug Fixes
    - Fixed DDR mode data sequence miss issue, which is caused by NIBBLE_POS.
  - Improvements
    - Increased g_sdmmc 512byte to improve the performance when application use a non-word align data buffer address.
    - Used OCR access mode bits to determine the mmccard high capacity flag.
- 2.2.3
  - Bug Fixes
    - Added response check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.1
  - Improvements
    - Improved MMC Boot feature.
- 2.2.0
  - Improvements
    - Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function.
    - Added strobe dll for mmc HS400 mode.
    - Added write complete wait operation for MMC_Write to fix command timeout issue.
- 2.1.2
  - Improvements
    - Improved SDMMC to support eMMC v5.0.
  - Bug Fixes
    - Fixed incorrect comparison between count and length in MMC_ReadBlocks/MMC_-WriteBlocks.
- 2.1.1
  - Bug Fixes
    - Fixed the block range boundary error when transferring data to MMC card.
- 2.1.0
  - Improvements
    - Optimized the function of setting maximum data bus width for MMC card.
- 2.0.0
  - Initial version

SD Card driver for MCUXpresso SDK

The current driver version is 2.4.2.

- 2.4.2
  - Improvements
    - Improved the erase timeout calculation logical in function SD_EraseBlocks according to SD specifications.
    - Added polling erase done status after each erase operations.
- 2.4.1
  - Improvements
    - Added macro SDMMCHOST_SUPPORT_VOLTAGE_CONTROL for the host which not support voltage control.
- 2.4.0
  - Improvements
    - Removed deprecated api in sd driver.
    - Added new api SD_PollingCardStatusBusy for application polling card status.
    - Improved the read/write/erase function flow.
    - Improved the signal line voltage switch flow.
    - Added powerOnDelayMS/powerOffDelayMS in sd_usr_param_t to allow redefine the default power on/off delay.
    - Added mutual exclusive access for init/deinit/read/write/erase function.
    - Fixed the driver strength configurations missed when timing mode switch to non SDR50/SDR104 mode.
    - Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.7, 10.4, 13.5, 14.4.
- 2.3.3
  - Improvements
    - Added host SDR timing mode capability validation during card initialization.
    - Added plling card ready for data status when transfer data failed.
    - Used cache line size align buffer for SD initialization api.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.3.2
  - Improvements
    - Moved power off function after card detect in SD_Init for DAT3 detect card feature.
- 2.3.1
  - Improvements
    - Removed the dead loop while polling DAT0 and CMD13 instead of using timeout mechanism.
- 2.3.0
  - Improvements
    - Marked api SD_HostReset/SD_PowerOnCard/SD_PowerOffCard/SD_WaitCardDetectStatus as deprecated.
    - Added new api SD_SetCardPower/SD_PollingCardDetectStatus/SD_HostDoReset.
    - Added internalBuffer in sd_card_t and removed rawCid/rawCsd/rawScr.

        * Added retuning support during data transfer under SDR50/SDR104 mode.
        * Increased the read/write blocks failed retry times for stability.
        * Added delay while retry the ACMD41 for stability.
- 2.2.12
  - Improvements
    * Increased the sd io driver strength for SD2.0 card.
  - Bug Fixes
    * Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).
- 2.2.10
  - Bug Fixes
    * Added event value check for all the FreeRTOS events to fix program hangs when a card event occurs before create.
- 2.2.7
  - Bug Fixes
    * Fixed MDK 66-D warning.
- 2.2.5
  - Improvements
    * Added SD_ReadStatus api to get 512bit SD status.
    * Added error log support in sdcard functions.
    * Added SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning and it is disabled by default.
- 2.2.4
  - Bug Fixes
    * Fixed DDR mode data sequence miss issue, which is caused by NIBBLE_POS.
  - Improvements
    * Increased g_sdmmc 512byte to improve the performance when application use a non-word align data buffer address.
    * Enabled auto cmd12 for SD read/write.
- 2.2.3
  - Bug Fixes
    * Added response check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.1
  - Improvements
    * Kept SD_Init function for forward compatibility.
- 2.2.0
  - Improvements
    * Separated the SD/MMC/SDIO init API to xxx_CardInit/xxx_HostInit.
    * SD_Init/SDIO_Init will be deprecated in the next version.
- 2.1.6
  - Improvements
    * Enhanced SD IO default driver strength.
- 2.1.5

  - Bug Fixes
    - Fixed Coverity issue.
    - Fixed SD v1.x card write fail issue. It was caused by the block length set error.
    - Fixed card cannot detect dynamically.
- 2.1.3
  - Bug Fixes
    - Fixed Non high-speed sdcard init fail at switch to high speed.
  - Improvements
    - Added Delay for SDCard power up.
- 2.1.2
  - Improvements
    - Improved SDMMC to support SD v3.0.
- 2.1.1
  - Bug Fixes
    - Fixed the bit mask error in the SD card switch to high speed function.
  - Improvements
    - Optimized the SD card initialization function.
- 2.1.0
  - Bug Fixes
    - Changed the callback mechanism when sending a command.
    - Fixed the performance low issue when transferring data.
  - Improvements
    - Changed the name of some error codes returned by internal function.
    - Merged all host related attributes to one structure.
- 2.0.0
  - Initial version.

SDIO Card driver for MCUXpresso SDK

The current driver version is 2.4.1.

- 2.4.1
  - Improvements
    - Added macro SDMMCHOST_SUPPORT_VOLTAGE_CONTROL for the host which not support voltage control.
- 2.4.0
  - Improvements
    - Removed deprecated api in sdio driver.
    - Improved the signal line voltage switch flow.
    - Added powerOnDelayMS/powerOffDelayMS in sdio_usr_param_t to allow redefine the default power on/off delay.
    - Added mutual exclusive access for init/deinit/direct/extend function.
    - Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.1, 12.2.
- 2.3.3

  - Bug Fixes
    - Fixed logical dead code coverity issue.
  - Improvements
    - Removed deprecated api in sdio driver.
- 2.3.2
  - Improvements
    - Added host SDR timing mode capability validation during card initialization.
    - Used cache line size align buffer for SDIO initialization api.
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.3.1
  - Improvements
    - Moved power off function after card detect in SD_Init for DAT3 detect card feature.
- 2.3.0
  - Improvements
    - Marked api SDIO_HostReset/SDIO_PowerOnCard/SDIO_PowerOffCard/SDIO_WaitCardDetectStatus as deprecated.
    - Added new api SDIO_SetCardPower/SDIO_PollingCardDetectStatus/SDIO_HostDoReset.
    - Added internalBuffer in sdio_card_t for card register content extract and improve the data access efficiency.
    - Added retry function after switch to target timing failed in SDIO_SelectBusTiming.
    - Changed defalut bus clock from 400KHZ to 25MHZ.
- 2.2.13
  - Improvements
    - Removed the sdio card interrupt from sdio host initialization, since the card interrupt enablement should be determined by application.
  - Bug Fixes
    - Fixed Out-of-bounds write Coverity issue.
- 2.2.12
  - Improvements
    - Added manual tuning function for looking for the tuning window automatically.
    - Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).
    - Fixed the fall through build warning by adding SUPPRESS_FALL_THROUGH_WARNING() in sdio driver.
- 2.2.11
  - Bug Fixes
    - Added check card async interrupt capability in function SDIO_GetCardCapability.
    - Fixed OUT OF BOUNDS access in function SDIO_IO_Transfer.
- 2.2.10
  - Bug Fixes
    - Fixed SDIO card driver get an incorrect io number when the card io number is bigger

than 2.
  - Improvements
    - Added SDIO 3.0 support.
    - Added API SDIO_IO_RW_Direct for direct read/write card register access.
- 2.2.9
  - Improvements
    - Added API SDIO_SetIOIRQHandler/SDIO_HandlePendingIOInterrupt to handle multi io pending IRQ.
- 2.2.8
  - Improvements
    - Updated sdmmc to support SDIO interrupt.
    - Added API SDIO_GetPendingInterrupt to get the pending io interrupt.
- 2.2.7
  - Bug Fixes
    - Fixed MDK 66-D warning.
- 2.2.6
  - Improvements
    - Added an unify transfer interface for SDIO.
  - Bug Fixes
    - Fixed Wrong pointer address used by SDMMCHOST_Init.
- 2.1.5
  - Improvements
    - Improved SDIO card init sequence and add retry option for SDIO_SwitchToHighSpeed function.
- 2.1.4
  - Improvements
    - Added Go_Idle function for SDIO card.
- 2.0.0
  - Initial version.

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

VGLite GPU Driver

The current version of the VGLite GPU Driver is 3.0.15_rev7.

- version 3.0.15_rev7
  - Fixed:
    - (MCUX-54842) Fixed build warnings
- version 3.0.15_rev6
  - Fixed:
    - Fixed incorrect scissoring issue in single thread mode
    - Optimized line stroking to reduce memory consumption
    - Extended blit output quality workaround to "vg_lite_blit_rect"
    - (IMX-3008) Fixed driver reporting incorrect version number
    - (IMX-2848) Allocated path stroking parameters dynamically
    - (IMX-3010) Fixed scissoring window check with large tessellation buffers
  - Changed:
    - (IMX-2907) Removed obsolete "vg_lite_perspective" API
- version 3.0.15_rev5
  - Fixed:
    - (IMX-2867) Fixed hang when processing vector paths with zero length
    - (IMX-2959) Fixed GPU using garbage data during image filtering
    - (IMX-2900) Restructured source code for better single thread & multithread modes maintenance
  - Changed:
    - (MCUX-52922) Disable GPU auto clock gating by default. Feature can be enabled from build config
- version 3.0.15_rev4
  - Changed:
    - (IMX-2900) Renamed build switch for disabling driver multithread support

- version 3.0.15_rev3
  - Fixed:
    - Relocated centerX/Y definitions in vg_lite.c
    - (IMX-2918) Reduced vg_lite_finish() delay when it has nothing to do
    - (IMX-2901) Fixed reversed red and blue channels in colour gradients fill colour
    - (IMX-2901) Fixed linear gradient matrix transformation error
    - (IMX-2901) Fixed radial gradient render error
  - Changed:
    - (IMX-2799) Enabled GPU auto clock gating by default
    - (IMX-2799) Added build switch to disable GPU auto clock gating
  - Added:
    - (IMX-2900) Added initial support for single thread mode
- version 3.0.15_rev2
  - Fixed:
    - (IMX-2881) Fixed memory leaks in vector path stroking implementation
    - (IMX-2863) Fixed stroked polygons rendering issue
    - (IMX-2842) Fixed system hang when drawing circular arcs
    - (MGG-897) Use OS heap instead of application heap for stroked vector polygons
    - (MGG-897) Use OS heap instead of application heap for ciecular arc rendering
  - Changed:
    - (IMX-2863) Allow users to configure fill colour for stroked & filled vector paths
- version 3.0.15_rev1
  - Fixed:
    - (IMX-2844) Fixed missing path descriptor initialization in "vg_lite_init_arc_path"
    - (IMX-2837) Fixed arc drawing direction
    - (IMX-2811) Added VGPE flush after buffer clear
  - Changed:
    - (IMX-2835) Optimized storage of radial gradients params to allow memory saving
  - Added:
    - Added dithering support for RT11xx platforms
    - Added color keying support for RT11xx platforms
    - (IMX-2817) Added vector path stroking
    - (IMX-2692) Added support for HW accelerated linear gradients on RT11xx platforms
- version 3.0.13_rev2
  - Fixed:
    - (MGG-793) Fixed clipping issue when using the RT500 blit output quality workaround
    - (MGG-830) Disabled RT500 blit output quality workaround for non-affine graphic transformations
    - (IMX-2701) Fixed memory leak in vector arc drawing API
    - (IMX-2699) Fixed build warnings in vector arc drawing API
    - (MGG-836) Fixed the font/text support via main VGLite driver API
  - Changed:
    - (IMX-1724) Changed image width 16 pixels alignment to stride 16 byte alignment
    - (MCUX-46210) Dropped useless "const" qualifier for the "name" attribute of "vg_lite_font_params_t" data structure

        - (MGG-836) Reordered "vg_lite_draw_text" API arguments
- version 3.0.13_rev1
  - Fixed:
    - (IMX-2577) Fixed support for colour palettes (CLUT) in multithread mode
    - (MGG-735) Fixed Elementary library instability caused by using calloc/free in ElmWrapBuffer
  - Changed:
    - (IMX-2600) Updated "vg_lite_finish" to wait for all frames previously submitted with "vg_lite_flush"
    - Aligned "vg_lite_radial_gradient_parameter" data struct with parameters in Elementary EVO object
  - Added:
    - Added support for drawing vector arcs/circles
    - Added support for i.MXRT6Q GPU
    - Added support for GCNanoliteV GPU Rev. 0x1322
    - Added vector arcs support in Elementary library
- version 3.0.11_rev3
  - Fixed:
    - Fix async event reset after being initialized
    - (IMX-2604) Fix polygon's rendering regression in multitasking scenarios
    - Avoid "vg_lite_blit" modifying user's transformation matrix
- version 3.0.11_rev2
  - Fixed:
    - (MGG-685) Added workaround to improve "blit" output quality for RT500
    - (MCUX-43004) Fixed clipping window regression issue introduced by VGLite 3.0.11.1
    - (MGG-764) Fixed VGLite heap useless splitting of memory nodes
    - (MGG-765) Fixed regression issue introduced by VGLite 3.0.11.1 when loading graphic resources using Elementary library
    - (IMX-2506) Fixed "vg_lite_update_rad_grad" not checking the result of memory allocation
    - (MCUX-42992) Fixed IAR toolchain not recognizing optimization directive
    - (MGG-763) Remove risk of out-of-bounds read in "vg_lite_update_rad_grad" function
  - Changed:
    - (IMX-2527) Improved memory footprint by using a common tessellation buffer for all drawing tasks
    - (MGG-712) Restructured OS abstraction layer to allow easier integration with popular OSes
- version 3.0.11_rev1
  - Fixed:
    - (IMX-2502) Fixed GPU command buffer overflow when copying context data
    - (IMX-2503) Fixed additional colour ring incorrectly appearing at the edge of radial gradients
    - (IMX-2487) Fixed risk of memory leak in "vg_lite_upload_path"
    - (IMX-2429) Fixed incorrect blending of A4 and A8 images (regression since VGLite 3.0.4.x)

    - (MGG-687) Fixed build warning when VG_RENDER_TEXT feature is disabled
  - Changed:
    - (IMX-2354) Added support for dynamic command buffer size management
  - Added:
    - (IMX-2435) Added new API function - vg_lite_get_transform_matrix - to calculate parameters for 2D perspective transformations
    - (IMX-2411) Added support for radial gradients in Elementary library
    - (IMX-2026) Added support for images embedded in EVO data in Elementary library
    - (IMX-2026) Added support for patterns embedded in EVO data in Elementary library
- version 3.0.9_rev2
  - Fixed:
    - (MCUX-40557) Fixed build warnings
- version 3.0.9_rev1
  - Fixed:
    - (MGG-648) Fixed rendered text overlapping issue
    - (MGG-650) Fixed memory leak caused by failure to unload RLE font data
    - (IMX-2395) Fixed incorrect reporting of indexed images as "supported" for GC355 GPU (RT1170)
  - Changed:
    - (IMX-2370) Refactored GPU driver HAL and OS layers
    - (MGG-646) Configured a vector font as default font
- version 3.0.9
  - Fixed:
    - (IMX-2361) Fixed tessellation bounds computation error
  - Changed:
    - (IMX-2367) Enabled alpha channel premultiplication by default for GC355 GPU (RT1170)
    - (IMX-2261) Added Elementary library input data address alignment verification
  - Added:
    - (IMX-2323) Added support for radial colour gradients for GC355 GPU (RT1170)
    - (IMX-2317) Upgraded the Elementary library to be thread safe
- version 3.0.6_rev4
  - Fixed:
    - (IMX-2357) Fixed rendering performance degradation since the implementation of the multithread/multicontext support
    - (MGG-576) Elementary: Fixed hard fault when resetting translation of EVO object
    - (MCUX-38672) Fixed font and text support build warnings
    - (MGG-596) Fixed memory leak in raster font loading
    - (MGG-596) Font and text support: Fixed out of range memory access in Elementary library
  - Changed:
    - (MGG-596) "VG_RENDER_TEXT=1" build symbol now required to enable font and text support
    - (MGG-594) Updated font and text support to allow easy decoupling from GPU driver and Elementary when not needed

        - (MGG-533) Removed "is_tspan" attribute from "vg_lite_font_attributes_t"
        - (MGG-533) Added new attribute "tspan_has_dx_dy" to "vg_lite_font_attributes_t"
        - (MGG-533) Added new argument "matrix" to "vg_lite_draw_text" API function
        - (MGG-592) Renamed "eFontTypes_t" enum to "eFontType_t"
        - (MGG-592) Renamed "eFontVectorType" identifier to "eFontTypeVector"
        - (MGG-592) Renamed "eFontRasterType" identifier to "eFontTypeRaster"
        - (MGG-596) Changed "vg_lite_draw_text" function return value from "int" to "vg_lite_error_t"
    - Added:
        - (MGG-596) Added "vg_lite_find_font" API function
        - (MGG-596) Added 2 new error codes for "vg_lite_error_t": VG_LITE_ALREADY_EXISTS and VG_LITE_NOT_ALIGNED
        - (IMX-2357) Allow users to override command queue task priority at build time using QUEUE_TASK_PRIO build symbol
        - (MGG-551) Added text wrapping support for vector fonts
        - (MGG-533) Added support for text right alignment
- version 3.0.6_rev3
    - Added:
        - (MGG-551) Added support for font and text rendering
- version 3.0.6_rev2
    - Fixed:
        - (IMX-2292) Fixed command buffer flushing after draw
        - (IMX-2293) Fixed copy of register status when command buffer was not full
        - (IMX-2305) Fixed scissor window taking no effect
        - (IMX-2324) Fixed GPU feature table reset when calling "vg_lite_close"
        - (IMX-2358) Fixed misuse of address operator in checking colour channel premultiplication flag
        - (MGG-542) Cleaned up useless "memset" in "vg_lite_init"
- version 3.0.6_rev1
    - Fixed:
        - (IMX-2295) Initialize task context to zero in vg_lite_init()
- version 3.0.6
    - Fixed:
        - (MGG-525) Fixed "vg_lite_init_path" not properly initializating the "path" data structure
    - Changed:
        - (IMX-2255) Updated "vg_lite_set_scissor" arguments to (x, y, width, height) instead of (x0, y0, x1, y1)
    - Added:
        - (IMX-2104) Added API to enable/disable colour channel pre-multiplication at runtime on RT1170
- version 3.0.5
    - Fixed:
        - (IMX-2252) Reset global mutex when it is destroyed
        - (IMX-2252) Fixed reset of task local context in vg_lite_close()

  - Changed:
    - (MGG-333) Enabled scissoring for GC255 GPU (i.MXRT500)
  - Added:
    - (IMX-1729) Added support for drawing from multiple threads
- version 3.0.4_rev5
  - Changed:
    - (IMX-2104) Disabled by default colour channel pre-multiplication on RT1170 platform
    - (MGG-517) Updated "vg_lite_draw_pattern" function to return VG_LITE_NOT_SUPPORT for A4/A8 patterns
  - Fixed:
    - (IMX-2155) Fixed hard coded image mode in "vg_lite_draw_pattern"
    - (IMX-2153) Updated "vg_lite_draw_pattern" to take into account pattern transparency
    - (KPSDK-37093) Elementary library - Fixed bad free in "load_evo"
    - (KPSDK-37093) Elementary library - Avoid resource leak in "ElmCreateBuffer"
- version 3.0.4_rev4
  - Fixed:
    - Fixed empty function argument lists definition for scissoring related API functions
    - (IMX-1995) Extended RT500 image rotation fix to vg_lite_blit_rect, vg_lite_draw_pattern
    - (IMX-1995) Isolated RT500 image rotation fix effects to RT500 platform only
- version 3.0.4_rev3
  - Fixed:
    - (IMX-1995) Compensated for RT500 image shift effect when rotation is approaching multiples of 90 dgs
- version 3.0.4_rev2
  - Fixed:
    - Fixed integration issue of "vg_lite_mem_avail" API
- version 3.0.4_rev1
  - Changed:
    - (IMX-1768) Enabled users to query, at runtime, the support for VG_LITE_UPPER draw quality
  - Fixed:
    - (IMX-2074) Fixed GPU exception handling issue
  - Added:
    - (IMX-2045) Added API to provide available heap memory
- version 3.0.4
  - Changed:
    - (IMX-1957) Enabled users to query, at runtime, the support for BORDER_CULLING and SCISSOR features
    - Enable users to query, at runtime, the support for RGBA 2 bits-per-channel image formats
  - Fixed:
    - (IMX-1934) Fixed image stride alignment verification for TILED images
    - Fix GC355 GPU (i.MXRT1170) draw error when tessellation window width is not aligned to 128

  - Added:
    - (MGG-333) Added support for GC355 GPU (i.MXRT1170) scissoring
- version 3.0.1_rev1
  - Fixed:
    - (MGG-250) Fixed GPU hang after a random time (mostly reproduced on RT1170 platforms)
    - (KPSDK-33132) Fixed Elementary library memory leaks in case of failed EBO loading
    - (MGG-336) Allow use of blend modes not affected by the border culling limitation
    - (MGG-18) Fixed Elementary library memory leaks when loading EVO/EBO/EGO objects
    - (MGG-353) Fixed linear colour gradient rendering error when loading EVOs using the Elementary library
- version 3.0.1
  - Changed:
    - Removed "vg_lite_blit2" API function due to lack of hardware support
    - Removed "vg_lite_scanline" API function due to lack of harware support
    - Aggregated "vg_lite_error.h" API header file content into "vg_lite.h"
    - Aggregated "vg_lite_features.h" API header file content into "vg_lite.h"
    - Aggregated "vg_lite_matrix.h" API header file content into "vg_lite.h"
    - Aggregated "vg_lite_path.h" API header file content into "vg_lite.h"
    - Aggregated "vg_lite_util.h" API header file content into "vg_lite.h"
    - (IMX-1861) Added return code to the "vg_lite_flush" API function
    - Changed VGLite GPU driver license from proprietary to MIT
  - Fixed:
    - Fixed definition of "elm_alloc" function in Elementary toolkit
    - (IMX-1869) Fixed initialization of aligned bytes in the command buffer
    - (IMX-1821) Fixed inverted background colours when using "vg_lite_draw_pattern"
    - Fixed hang when calling "vg_lite_flush" repeatedly
    - (IMX-1861) Fix propagation of return codes from "stall", "submit", "vg_lite_flush" function calls
- version 2.0.14_rev1
  - Changed:
    - (IMX-1809) Fixed misspelling of "vg_lite_buffer_transparency_mode"
    - (IMX-1778) Added verification of colour gradients parameters
    - (IMX-1813) Added return code to the "vg_lite_hal_allocate_contiguous" function
    - (MGG-204) Added return code to "vg_lite_finish"
  - Fixed:
    - (IMX-1808) Fixed "vg_lite_blit" failure on dynamically allocated buffers
    - (IMX-1773) Fixed failure to create 16 colours gradients
    - (IMX-1790) Fixed driver incorrectly reporting available heap space
    - (IMX-1810) Fixed verification of raster image stride alignment
    - (IMX-1810) Fixed verification of raster image colour depth
    - (IMX-1816) Fixed "vg_lite_close" not releasing memory allocated from OS heap
    - (MGG-201) Fixed hard fault caused by command buffer management
    - (MGG-202) Fixed "vg_lite_hal_wait_interrupt" function ignoring the timeout

        * (MGG-203) Fixed "vg_lite_draw" function always returning success
- version 2.0.13_rev2
  - Fixed:
    * (MGG-102) Fixed incorrect colour gradient clipping issue when using "vg_lite_draw_-gradient" API
    * (MGG-140) Fixed "vg_lite_draw_gradient" error when gradient is not covering the entire shape
- version 2.0.13_rev1
  - Added:
    * (MGG-88) Support for operating with BGRA2222, ABGR2222, ARGB2222 type images
    * (MGG-88) Support for operating with ABGR4444, ARGB4444 type images
    * (MGG-88) Support for operating with ABGR8888, ARGB8888 type images
    * (MGG-88) Support for operating with XBGR8888, XRGB8888 type images
    * (MGG-52) Improved GPU bus error reporting by using weak functions
  - Changed:
    * (MGG-66) Restructured GPU driver by exposing the HAL source code for easier integration with operating systems
  - Fixed:
    * (MGG-72) Fixed rough edges of vector artefacts when using the "vg_lite_draw_pattern" API
    * (MGG-58) Fixed "vg_lite_blit_rect" API not supporting a zero Y coordinate
- version 2.0.13_rev0

NXP WiFi

Version 1.3.r47.p16

- Updates:
  - Updated FC's f/w version to 2.p66.155.
  - Updated CA2 and RB3+ f/w version to 21.p124.
  - Fine tuning of GTK rekey offload feature.

Version 1.3.r47.p15

- Bug Fix:
  - DUT fails to connect to Ex-AP configured with wpa2 Enterprise security (Auth method Fast-mschapv2).
  - Incorrect AKM types PSK(2), PSK(SHA-256) are seen in beacon after configuring APUT in wpa2-psk security mode.
  - Link lost seen after wlan-scan when DUT is connected in wpa2 ft-psk security in 2.4Ghz band.

Version 1.3.r47.p12

- Bug Fix:
  - Throughput numbers for TCP-Rx traffic on APUT are dropping to 0mbps in HE20/VHT20

mode.

Version 1.3.r47.p11

- Bug Fix:
  - Failed to connect STAUT configured in WPA2+PMF required security to Ex-AP configured in WPA2/WPA3 mixed mode+PMF capable security.
  - STAUT is waking up with Broadcast traffic while running MEF AUTO PING ALLOW and WAKE HOST & MEF AUTO PING DISCARD and WAKE HOST.
  - Cannot connec to AP with security wpa3_sb_192_eap_tls.
  - Cannot out band independent reset successfully.

Version 1.3.r47.p10

- Bug Fix:
  - Auto reconnect - link lost reported when ap shut down.
  - STAUT failed to roam from Ex-AP1 to Ex-AP2 with WPA3 security mode while running UDP-Tx traffic.
  - STAUT is not waking up from suspend state while running Host-sleep test even after running Unicast/Broadcast traffic from Ex-AP to STAUT.
  - STAUT is waking up with Unicast data and without unicast data as well before running Broadcast data traffic from AP backend to STAUT.
  - While running WPS connection with Ex-AP, DUT gets deauthenticated after M4 packet, WPS_NACK is recvd. M1 packet does not have WPS PBC or keypad specific bit set
  - "Network not found" is seen on the STAUT after disconnection and reconnection of the same profile in UNII4 channels (169,173,177).
  - When Dut reported Link lost, After configured time interval, DUT reconnects to AP but fails to ping to AP_BACKEND after reconnection.
  - Coverity Fixes.

Version 1.3.r47.p9

- Bug Fix:
  - APUT is advertising channel width information for 40Mhz in HE-phy capabilities instead of 20Mhz bandwidth in Assoc response frame.
  - STAUT failed to roam from Ex-AP1 to Ex-AP2 with WPA3 security mode with same channel same band and different channel same band scenario.
  - 15-20% Throughput degradation observed in TCP-TX and UDP-TX of HE[40Mhz/80MHz], VHT[40Mhz/80Mhz] compared to previous release[R45.p12].
  - "WLAN: Network not found" is observed while connecting STAUT configured in wpa2 security PMF capable to Ex-AP in wpa2 security with PMF required.
  - Coverity Fixes.

Version 1.3.r47.p8

- Bug Fix:
  - "WLAN: network not found & Warn: Scan temporary failure" is observed after disconnecting from one network and Re-connecting to same network after renaming SSID of Ex-AP.
  - "Network not found" is seen on the STAUT after disconnection and reconnection of the same

profile in UNII4 channels (169,173,177).
  - Beacons not stopping in the older channel, after uAP switching the channel according to the Ex-AP in simultaneous mode.
  - Incorrect return value on error.

Version 1.3.r47.p7

- Bug Fix:
  - STA STRESS | Independent Reset, DUT went to hang state after 134 iterations of independent reset with status "ASSERT: wlan_process_hang: 982 Assert failed: Panic"!
  - Error message "'is_mef' undeclared(first use in this function)" is observed while the time of compilation of binary after enabling macro for the host-sleep.
  - DUT not able to roam from AP1 to AP2 in different channel (DFS) and same Band on reducing the RSSI of AP1.
  - STAUT failed to roam from Ex-AP1 to Ex-AP2 with WPA3 security mode while running UDP-Tx traffic.
  - MEF, DUT fails to wakeup with mef conditions ping 1,3 and arp 1 & 3.

Version 1.3.r47.p6

- Features
  - Add GTK rekey offload support.
  - Independent Reset via In-band
  - Independent Reset via Out-of-Band
- Updates:
  - Updated FC's f/w version to 2.p66.14.
- Bug Fix:
  - Power save/host sleep support improvements.
  - Messy log output after in band independent reset(17 in 31)
  - Association_Req content corrupted when Dut config as WPA/WPA2 mix mode and connecting to AP WPA+TKIP
  - Cannot connect to AP with security wpa3_sb_192_eap_tls.
  - Command "wlan-add-packet-filter 1" is not available for setting MEF filter configuration in Host-sleep.
  - WPA3 Enterprise support (Host based - Remaining set of authentication methods)
  - STAUT failed to roam from Ex-AP1 to Ex-AP2 with WPA3 security mode while running UDP-Tx traffic.
  - No Link lost is observed while switching from 11n[2.4G | 20Mhz] to legacy[2.4G] mode.
  - The introduction of new features does not follow the scope limitations of modules and apps
  - STAUT 11r, DUT unable to roam to particular BSSID after 10 to 15 iterations of roam, showing "Roaming already in progress Started FT roaming"
  - STAUT | After DUT wake up from Hostsleep condition 0x10 (WAKE_ON_ARP_BROADCAST) Ping failed from AP_Backend to DUT and Delayed ping from DUT to AP_Backend
  - Error: ignoring scan request in invalid state
  - Fix COVERITY Issue for midware_wifi

Version 1.3.r47.p5

- Updates:
  - Updated FC's f/w version to 2.p66.11.
- Bug Fix:
  - AKM checks are wrong for UAP PMF MANDATORY WPA2 PSK STA PMF MANDATORY WPA2 PSK for 11AN[20Mhz] as well as BGN[20Mhz] mode.
  - STAUT is not waking up in any scenarios after giving Host-sleep commands with default power save enabled mode.

Version 1.3.r47.p4

- Features
  - Added support for Doxygen.
- Bug Fix:
  - AKM checks are wrong for UAP PMF MANDATORY WPA2 PSK STA PMF MANDATORY WPA2 PSK for 11AN[20Mhz] as well as BGN[20Mhz] mode.
  - On DUT manually configured parameters of 11axcfg, not getting reset to default after "wlan-reset" command execution.
  - 2G to 5G and 5G to 2G FTOA roaming fails , link lost observed with permanent disconnect
  - Change of bandwidth from 80Mhz to 40 MHz is observed after uAP (APUT) is stopped and started again, in HE/VHT band.
  - STAUT unable to scan Ex-AP in UNII4 channels (169,173,177) with any Band/security.
  - "Command response timed out. command 0x107" observed after 945 iterations in Scan--Connect-Disconnect Test without any traffic [Idle Test].
  - RT Console hangs after enabling WLAN/BT Independent Download.
  - Cannot get full iperf summary.
  - While running UDP Bi-directional traffic, QOS data packets are not going from APUT/STAUT.

Version 1.3.r47.p3

- Updates:
  - Updated FC's f/w version to 2.p66.6.
  - Updated CA2 and RB3+ f/w version to 21.221.
- Features
  - Added FW (Parallel) Download support.
- Bug Fix:
  - WiFi hang up with memory alloc buffer error during perform WiFi-scan along with WiFi independent reset loop test(OT already form the NTW)
  - TCP-Keep-Alive packets are not seen in sniffer after successfully setting configuration commands for cloud keep alive and STAUT is not waking up after putting in suspend state.
  - STAUT is not waking up from suspend state while running Host-sleep test even after running Unicast/Broadcast traffic from Ex-AP to STAUT.
  - AKM checks are wrong for UAP PMF MANDATORY WPA2 PSK STA PMF MANDATORY WPA2 PSK for 11AN[20Mhz] as well as BGN[20Mhz] mode.
  - Ex-STA (Kestrel, Firecrest RT1060-EVKC) Failed to associate in WPA2 PMF disabled mode to APUT configured in WPA2/WPA3 mixed security mode.
  - STAUT, failed to Associate in WPA2 security with PMF disabled to Ex-AP configured in

WPA2/WPA3 mixed security mode.

Version 1.3.r47.p2

- Bug Fix:
  - Fail to disable IEEE Power save mode, while giving command to disable it.
  - NXP devices Kestrel/KF2-RD(STA) unable to connect to DUT as MMH-AP but connecting with Mobile, Four way Handshake Timeout is observed with NXP Devices
  - Ex-STA(Kestrel, KF2) Assoc-reject is observed with uAP configured in OWE security mode.
  - Cannot connect to 5g external AP.

Version 1.3.r47.p1

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p109
  - Updated 8801 f/w version to 36_186
  - Updated Firecrest firmware version to p66.5.
- Features
  - Added EVKC board support for all SoCs.
  - Added WPA2/3 Enterprise support for EAP-SIM/EAP-AKA/EAP-AKA-PRIME for STA and uAP.
  - Added CSI support.
  - Added Auto Reconnect support.
  - Added CA2 and RB3+ Parallel FW Download support.
  - Added CA2 and RB3+ Independent Reset via In-band support.
  - Added CA2 and RB3+ Independent Reset via Out-of-Band support.
  - Added CA2 and RB3+ Boot sleep patch support.

Version 1.3.r46.p7

- Updates:
  - Updated Falcon to p185, CA2 and RB3+ f/w version to 21.p91.5
- Bug Fix:
  - uAP reassociation not working
  - Traffic stops wen moving from auto rate to mcs0
  - DUT not sending TCP-KEEP-ALIVE in suspend state
  - DUT not able to roam from WPA-R1 enabled AP to WPA3-R3 enabled AP
  - Ex-AP1 to Ex-AP2 roaming not working in wpa3
  - STAUT is nto following BSS transition correctly
  - RF test mode commands crash issue
  - Ping is not working when uAP configured with wpa3-sb-192 bit EAP-TLS/EAP-TTLS/EAP-PEAP enterprise security.
  - DUT shows network not found, even when connected to Ex-AP
  - Stress test fixes

Version 1.3.r46.p5

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p91.5

- Bug Fix:
  - Wifi random crash issue when DUT set to sleep
  - Enable 11D for uAP by default
  - STA doesnt get IP address when ieee-ps and deep sleep are enabled
  - Coverity fixes
  - BT/BLE fix for PTS test case

Version 1.3.r46.p4

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p91.4
- Bug Fix:
  - Fixed: pre-cert: 20/40 BSS coexistence management is not supported in association request.
  - Fixed: pre-cert: STAUT is not correctly following the MU EDCA parameters advertised by the AP.
  - Fixed: Fail to connect to external AP with security wpa3.
  - Fixed: uAP with wpa3 security is not connectable.
  - Fixed: DUT not able to connect to Ex-AP in WPA2 security when PMF required is set.
  - Fixed: Remove BAND_B rates from Supported rates for 5G channels during scan.
  - Fixed: pre-cert STAUT is not including operating class 81 and 115 under Alternate Operating Classes in assoc request.
  - Fixed: Cannot connect to external AP successfully.
  - Fixed: 5Ghz channel are configured acceptable in 2.4Ghz band
  - Fixed: wlan_set_rf_tx_power is not consistent with tx-frame power output
  - Fixed: Getting compilation errors while compiling the binary after defining "CONFIG_UNI-I4_BAND_SUPPORT" macro in wifi_config.h file.
  - Fixed: uAP Beacons advertise 3 AKM suites (PSK & PSK SHA256 & SAE SHA256) in RSNIE when WPA2+WPA3 mixed mode is set on AP with mfpc 1 and mfpr 0.
  - Fixed: STAUT configured with WPA3-SAE associates to WPA2 configured AP.
  - Fixed: DUT unable to roam from Ex-AP1 to Ex-AP2 on lowering the RSSI of Ex-AP1.
  - Fixed: pre-cert STAUT is not correctly following the MU EDCA parameters advertised by the AP.
  - Fixed: DUT-STA is unable to associate with WPA/WPA2 Mixed mode AP when DUT-STA is configure WPA security.
  - Fixed: pre-cert STAUT fails to roam from WPA3 configured AP1 to WPA2-PSK configured AP2.
  - Removed IEEE_MGMT_ACTION_CATEGORY_PUBLIC enum.
  - Fixed logic for consecutive connect for both embedded and wpa supplicant.
  - Fixed all compilation errors and warnings seen on dapeng.
  - Resolved MISRA and coverity defects.

Version 1.3.r46.p3

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p91.2
- Bug Fix:
  - Fixed: TCP and UDP TX traffic stream not working with ex-sta

  - Fixed: pre-cert: STAUT is not following BSS transition correctly
  - Fixed: Extend wlan-get-antcfg to print current antenna
  - Fixed: DUT able to configure MFPR 0 in WPA3 security.
  - Fixed: DUT MCS rates are not updating to default after stopping/removing previous profile with MCS set to fixed number via fixed rate command.
  - Fixed: DUT-uAP unable to start with ACS configuration.
  - Fixed: While running WPS connection with Ex-AP, DUT gets deauthenticated after M4 packet, WPS_NACK is recvd. M1 packet does not have WPS PBC or keypad specific bit set.
  - Fixed: Fail to create ipv6 iperf client.
  - Fixed: uAP beacons contain incorrect bit, UAPSD bit is set in WME QoS Info even though it is not supported.
  - Fixed: uAP beacons contain incorrect bits, SU/MU Beamformer bits are set in HE Phy capabilities and UAPSD bit is set in WME QoS Info even though it is not supported.
  - Fixed: Data path blocks once we initiate wlan-scan on DUT and change Ex-AP channel at the same time.
  - Fixed: MISRA defects.

Version 1.3.r46.p2

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p91
- Bug Fix:
  - Fixed BT connection issue on RB3+

Version 1.3.r46.p1

- Updates:
  - Updated CA2 and RB3+ f/w version to 21.p90
  - Updated 8801 f/w version to 36_181
  - CA2 Fixed FIPS GCMP support
  - Added all changes as part of r45.p3 as applicable for CA2 and RB3+
  - Added information for BSD3 license.
  - Updated WLCMGR and Wi-Fi driver to support WPA supplicant and hostapd.
  - Deepsleep feature in BLE peripheral role
  - Deep Sleep and wakeup feature is enabled for CA2 and RB3+
- Features
  - Added WPA supplicant and hostapd support.
  - Added WPS 2.0 support for STA and uAP.
  - Added WPA2 enterprise support for STA and uAP.
  - Added WPA3 enterprise support with suite b and suite b 192 bit mode for STA and uAP.
- Bug Fixes:
  - Fixed: APUT goes into hang state after every disassociation of STA.
  - Fixed: (pre-cert)AMSDU Rx test fails as STAUT returns the throughput received as 0.
  - Fixed: Country Information IE with default country code (WW) is seen in beacon even after changing the regions from WW to US/EU/CA/CN.
  - Fixed: DUT should show "Network not found" message in cli, when Configure SSID is not

present in network area.
- Fixed: DUT not able to connect to Ex-AP in WPA3R3 security.
- Fixed: DUT not able to roam from Ex-AP1 to Ex-AP2.
- Fixed: Hostsleep/MEF test condition failures are seen.
- Fixed: uAP not coming up in WPA2/WPA3 mixed mode security., getting "WPA initialization failed" on DUT console.
- Fixed: DUT shows wrong channel number in wlan-scan-opt and wlan-info command results.
- Fixed: DUT sending QoS data packets with LDPC coding when LDPC coding is disabled on Ex-STA.
- Fixed: Country Information IE is not seen in uAP Beacon.
- Fixed: DUT is going into hang state while running Connect/Disconnect Stress Test
- Fixed: Supported MCS and NSS rates are not properly set under HE capabilities IE in beacon frame when CAPA enabled binary is flashed on DUT.
- Fixed: Channel Bandwidth is setting to 80Mhz when uAP is configured in 11AX-40Mhz/20-Mhz.
- Fixed: Beacon frame includes HE IE when uAP is configured in 11AC using Capa command.
- Fixed: DUT is not connecting in WPA2/WPA3 Enterprise security.
- Fixed: DUT is not getting ipv4 address after Roaming to Ex-AP2.
- Fixed: DUT is not roaming from AP1 to Ex-AP2 after running wlan-ft-roam command in 5Ghz.
- Fixed: DUT is not following BSS transition Correctly.
- Fixed: DUT is not responding Beacon Report Action frame after sending the Beacon request from Ex-AP to DUT.
- Fixed: Automatic Channel Selection is not working when DUT configured in MMH mode.
- Fixed: DUT is going for Reassociation with the same Ex-AP after running "wlan-host-11k-neighbour-req" command.
- Fixed: uAP Beacons does not include SHA256 in RSNIE when PMF mandatory is set on AP in WPA2-PSK security.
- Fixed: Low Throughput seen in 11AX, 11AC, 11N (80MHz, 40MHz & 20MHz) TCP-RX/T-X, UDP-TX/RX in 2.4Ghz & 5Ghz.
- Fixed: DUT is not using "FT using PSK" AKM suite when 802.11R is enabled.
- Fixed: DUT not able to connect to Ex-AP in WPA2 security when PMF required is set.
- Fixed: Need to reset DUT every time for every new connection to happen.
- Fixed: Incorrect VHT IE "RX/TX MCS Map" is seen in uAP beacons.
- Fixed: uAP Beacons is advertising 4 Pairwise Cipher Suites [CCMP (256), GCMP (256), AES (CCM), GCMP (128)] in RSNIE with WPA2/WPA3 security.
- Fixed: uAP not coming up in 2.4GHz channel 11, getting "uAP start failed, giving up" on DUT console.
- Fixed: DUT is going in hang state after Stop/Remove/Start network in wpa2 and wpa3 security, when configured in MMH mode.
- Fixed: HE IE is not reflecting in beacons, when DUT is configured in default mode without capa in MMH.
- Fixed: DUT is not connecting in WPA3 security, shows "Error: Init of random number generator failed."

Version 1.3.r45.p12

- Updates:
  - Uodated tx pwr limit files for murata 2el module.
  - Added config macro for RU Tx power.
  - Updated WPA2/3 Enterprise support to handle pre-cert tests.
- Bug Fixes:
  - Fixed: uAP not starting up in channel 14 showing start failed when uAP country code is set to JP.

Version 1.3.r45.p11

- Updates:
  - Added integrate tx pwr limit files for murata 2el module.
  - Added support for legacy mode of RU Tx power.
  - Updated WPA2/3 Enterprise support to handle pre-cert tests.
- Bug Fixes:
  - Fixed: Bi-directional traffic converts to uni-directional traffic.
  - Fixed: The support for "wlan-multi-mef" command to configure MEF parameters on cli is not available.
  - Fixed: Country Information IE with default country code (WW) is seen in uAP beacon even after changing the regions from WW to US/EU/CA/CN.
  - Fixed: Failed to wakeup card after turned on IEEE Power Save mode.
  - Fixed: Bi-directional iperf traffic is not running getting dropped to 0.000 bits/sec.
  - Fixed: Macbook not connecting with DUT-AP in channel 48 and BW 80MHz.
  - Fixed: [pre-cert] DUT fails to connect to Ex-AP configured with wpa/wpa2 enterprise security (AES Encryption) and the radius server configured with hostapd, unsupported certificate error.
  - Fixed: Not connecting to AP in WPA-TKIP and WPA-AES, M2 is not being initiated by STA.

Version 1.3.r45.p10

- Updates:
  - Updated firmware version to 2.p7.19 and added TP signed FW(with VDLL) too.
  - Changed the init and command flow for uart_wifi_bridge app.
- Bug Fixes:
  - Fixed: [pre-cert] DUT not correctly receiving AMPDU+AMSDU , AMSDU bit is not set to 1 in QOS data frame.
  - Fixed: uAP not starting up in channel 14 showing start failed when uAP country code is set to JP.
  - Fixed: Data path blocks once we initiate wlan-scan on DUT and change Ex-AP channel at the same time.
  - Fixed: App will block/output error log when trying to set rf tx frame.

Version 1.3.r45.p9

- Updates:
  - Updated firmware version to 2.p7.17 and added TP signed FW(with VDLL) too.
  - Added support for channel based RU Tx power.

  - Added support of reassociate command on STAUT to test reassociate feature.
- Bug Fixes:
  - Fixed: Ex-STA(Kestrel) not able to connect to uAP with "reassociate" command in first attempt, getting deauthentication and again connection is initiated with Association Request and connection happens, in wpa3 security.
  - Fixed: [pre-cert] STAUT is sending incomplete beacon report response to AP's beacon report request.
  - Fixed: Messy log output when trying to turn off deep sleep mode.
  - Fixed: Unable to set TX-OMI on uAP using command "wlan-set-tx-omi".
  - Fixed: DUT is not re-connecting when bandwidth/mode changed on Ex-AP.

Version 1.3.r45.p7

- Updates:
  - Updated firmware version to 2.p7.15 and added TP signed FW(with VDLL) too.
- Bug Fixes:
  - Fixed: STAUT is not following BSS transition correctly.
  - Fixed: DUT is not sending TCP-KEEP-ALIVE packets in suspend state, when TCP connection is established via Cloud keep alive command.
  - Fixed: DUT not able to roam from Ex-AP1 to Ex-AP2 on lowering the RSSI of Ex-AP1 in wpa3 security.
  - Fixed: STAUT's probe request does not contain MBO-OCE IE
  - Fixed: Ping is not working when uAP configured with wpa3-sb-192 bit EAP-TLS/EAP-TTLS/EAP-PEAP enterprise security.
  - Fixed: STAUT's probe request does not contain MBO-OCE IE.
  - Fixed: Throughput enhancement for STA and uAP mode for various TCP/UDP and Tx-Rx modes for embedded and wpa supplicant.
  - Fixed: "TCP_ABORTED_LOCAL" message seen on DUT console instead of "TCP_DONE" when running iperf traffic
  - Fixed: [pre-cert] STAUT is not dropping TP when AP is increasing MPDU spacing factor from 0 to 3.

Version 1.3.r45.p6

- Updates:
  - Updated firmware version to 2.p7.11 and added TP signed FW(with VDLL) too.
  - uart_wifi_bridge added and errors fixed for RT1170-EVKB.
- Bug Fixes:
  - Fixed: Traffic stops after moving from auto rate to fixed mcs0. Command is triggered on the fly.
  - Fixed: DUT not able to roam from WPA3R1 enabled AP to WPA3R3 enabled AP showing network not found.

Version 1.3.r45.p5

- Updates:
  - Updated firmware version to 2.p7.10 and added TP signed FW(with VDLL) too.
- Bug Fixes:

  - Fixed: Data path blocks once we initiate wlan-scan on DUT and change Ex-AP channel at the same time.
  - Fixed: uAP beacons contain incorrect bits, SU/MU Beamformer bits are set in HE Phy capabilities and UAPSD bit is set in WME QoS Info even though it is not supported.
  - Fixed: uAP beacons contain incorrect bit, UAPSD bit is set in WME QoS Info even though it is not supported.
  - Fixed: [pre-cert] STAUT is not correctly following the MU EDCA parameters advertised by the AP
  - Fixed: [pre-cert] 20/40 BSS coexistence management is not supported in association request.
  - Fixed: Fail to connect to external AP with security wpa3.
  - Fixed: uAP with wpa3 security is not connectable.
  - Fixed: DUT not able to connect to Ex-AP in WPA2 security when PMF required is set.
  - Fixed: Remove BAND_B rates from Supported rates for 5G channels during scan.
  - Fixed: Fix wifi_cli_prov example hang when connect if enable ieee power save and deep sleep.
  - Fixed: RFTM: 5Ghz channel are configured acceptable in 2.4Ghz band.
  - Fixed: RFTM: wlan_set_rf_tx_power is not consistent with tx-frame power output.
  - Fixed: DUT not able to roam from AP1 to AP2 in different channel (DFS) and same Band on reducing the RSSI of AP1.
  - Fixed: uAP Beacons advertise 3 AKM suites (PSK & PSK SHA256 & SAE SHA256) in RSNIE when WPA2+WPA3 mixed mode is set on AP with mfpc 1 and mfpr 0.
  - Fixed: DUT unable to roam from Ex-AP1 to Ex-AP2 on lowering the RSSI of Ex-AP1.
  - Fixed: Fixed build errors and warnings for matter, MISRA, coverity and other compilers.

Version 1.3.r45.p4

- Updates:
  - Updated firmware version to 2.p7.4 and added TP signed FW(with VDLL) too.
- Bug Fixes:
  - Fixed: Fail to create ipv6 iperf client.
  - Fixed: DUT MCS rates are not updating to default after stopping/removing previous profile with MCS set to fixed number via fixed rate command.
  - Fixed: MISRA Defect fixed.
  - Fixed: IEEE PS event not getting triggered.
  - Fixed: While running WPS connection with Ex-AP, DUT gets deauthenticated after M4 packet, WPS_NACK is recvd. M1 packet does not have WPS PBC or keypad specific bit set.
  - Fixed: Fail to create ipv6 iperf client.
  - Fixed: Hang issue seen when assoc reject event received.

Version 1.3.r45.p3

- Updates:
  - Enabled h/w acceleration APIs via mbedtls.
  - VDLL support added.
- Bug Fixes:
  - Fixed: (pre-cert)Need support to set UL MU Disable/Data Disable element through TX-OMI

command.
  - Fixed: iTCP and UDP TX traffic stream not working with ex-sta.
  - Fixed: STA not able to connect to APUT when APUT is configured in OWE security.
  - Fixed: DUT is going on hang state after running "wlan-reset" command.
  - Fixed: "TCP_ABORTED_REMOTE" message seen on DUT console when running UDP traffic.
  - Fixed: (pre-cert)STAUT is not scanning Non Transmitted SSID in MBSSID test.
  - Fixed: 6+ ms of traffic burst is seen which is failing cert criteria of max cot as 6ms.
  - Fixed: DUT not getting IP address(DHCP) when "WMM_ENH" Macro is defined in "wifi_config.h" file for TWT feature.
  - Fixed: For Tx Frame & tx-continuous, the Measured tx-power value is 2dBm irrespective of configured Tx-power(10/12/15/20dBm)
  - Fixed: Data path blocks once we initiate wlan-scan on DUT and change Ex-AP channel at the same time.
  - Fixed: DUT not able to scan and connect to DFS channels when Broadcast SSID is disabled.
  - Fixed: Aggregation is broken when TCP/UDP TX Data is going with TID 6-7 (Voice), and TID 3 (Best Effort).
  - Fixed: uAP not beaconing in 40MHz when uAP configured in channel 12/13-40MHz in MMH mode
  - Fixed: STA Ping is not working once it roams to Ex-AP2.
  - Fixed: Wi-Fi lwip port does not protect correctly concurrent accesses to lwip stack.
  - Fixed: Beacon frame does not include RSN XE IE when uAP configured in WPA3R3 security.
  - Fixed: Beacon frame includes HT IE when uAP configured in 11A mode using capa legacy command.
  - Fixed: DUT not connecting to Ex-AP when PMF is configured using "wlan-set-pmfcfg" command.
  - Resolved IAR, MCUX and MDK build errors.

Version 1.3.r45.p2

- Updates:
  - Updated firmware version to 2.p7.1 and added TP signed FW too.
  - Added RFTM commands missing CLI commands related to OFDMA feature.
  - Added support for EVKB board.
- Bug Fixes:
  - Fixed: uAP not coming up in channel 14 when country code is set to JP.
  - Fixed: DUT not able to connect with 32-character SSID when Ex-AP configured in open/wpa2/wpa3 security.
  - Fixed: QoS Data packets not seen on air when we run "wlan-set-tx-omi 0x48 0xff 16" command on DUT.
  - Fixed: [pre-cert] Not able to set non-preferred channel in MBO through wpa_cli on RTOS.
  - Fixed: Automatic Channel Selection is not working when DUT configured in MMH mode.
  - Fixed: STA not able to connect to APUT when APUT is configured in OWE security.
  - Fixed: Wi-Fi lwip port does not protect correctly concurrent accesses to lwip stack.
  - Fixed: DUT is going on hang state after running "wlan-reset" command.
  - Resolved IAR, MCUX and MDK build errors.

Version 1.3.r43.p9

- Updates:
  - Added support for RF test mode.
- Bug Fixes:
  - Fixed: (pre-cert)Updated help message for TX-OMI command.
  - Fixed: (pre-cert)STAUT is not scanning Non Transmitted SSID in MBSSID test.
  - Fixed: (pre-cert)STAUT is not governing OBSS Narrow Bandwidth RU in UL OFDMA Tolerance Support.
  - Fixed compilation errors and warnings after branch merge.

Version 1.3.r44.p3

- Bug fixes:
  - Added RSNX IE for WPA3-R3 for uAP

Version 1.3.r44.p2

- Updates:
  - Updated CA2, RB3+ firmware to p82
  - Bug fixes:
    - Fixes for 11KR
    - Fixes for WPA3-R3 STA
    - Added extra event for Connected notification in case of uAP

Version 1.3.r44.p1

- Updates:
  - Updated CA2, RB3+ firmware to p79
  - Updated 8801 firmware to p180
- Features:
  - Added 11KVR support for CA2 and RB3+
- Bug fixes:
  - Fixed: Association Request does not include Group Management Cipher Suite IE in RSNIE when PMF mandatory is set on STA in WPA2-PSK/WPA3/OWE security.
  - Fixed: Sending of deauth codes in all scenarios
  - Fixed: Command timeout issues for wlan-eu-crypto
  - Added Coverity fixes
  - Fixed command timeout seen during simultaneous AP + STA mode, with high traffic on 8801
  - Set proper value for Edmac value updated to support Murata 2DS Module
  - Fixed Channel switch announcement is not seen in beacon in AP+STA mode, when AP changes channel on 8801
  - Added few BT Sig qualification fixes

Version 1.3.r43.p8

- Updates:
  - Updated firmware version to p182.1 and added TP signed secure FW too.
- Bug Fixes:
  - Fixed: (pre-cert)APUT beacons does not have RSNXE when configured in H2E mode.

Version 1.3.r43.p7

- Updates:
  - Updated firmware version to p182.
  - Added TP signed secure FW.
  - Added support for 2EL M2 module for Firecrest.
  - Added wifi capability configuration support.
  - Added FIPS validation feature.
  - Added wifi rf test mode example.
  - Unified all wifi examples in wifi_cli.
  - 11R is not surrpoted for FC.
- Bug Fixes:
  - Modified the method to load RU tx power limit with RT.
  - Modified txrate config design for 11AC and 11AX features.
  - Added support for roaming and neighbor request processing.
  - Fixed: "Error in sending Background traffic" messages seen on DUT while running Background TX-Traffic.
  - Fixed: STAUT is not associating in WPA3 Hash-to-Element only mode.
  - Fixed: Not able to change HE MAC Capabilities or HE PHY Capabilities in Association Request using 11axcfg command.
  - Fixed: DUT not waking up from suspend state when we start multicast traffic after 30sec-1 minute.
  - Fixed: STAUT hangs after scanning AP in scenario K of SI-5.2.2 test case.
  - Fixed: STAUT is not including operating class 81 and 115 under Alternate Operating Classes in assoc request.
  - Fixed: STAUT fails to roam from WPA3 cionfigured AP1 to WPA2-PSK configured AP2.
  - Fixed: DUT goes for DHCP DORA Process after roaming to AP2 from AP1.
  - Fixed build errors and warnings.
  - Corrected description related to num_dat_pkts parameter of tx-omi command.

Version 1.3.r43.p6

- Updates:
  - Updated firmware version to p174 and added latest signed FW bins as well.
- Bug Fixes:
  - Fixed: Command timeout issues observed for various offload commands.
  - Fixed: TWT requester bit not set in Extended Capabilities IE (127) in Association Request and Probe Request of STAUT.
  - Fixed: Not able to scan and connect Firecrest-STA to Firecrest-uAP configured in UNII-4 band channels 173 and 177.
  - Fixed: Multiple error messages are seen on DUT while running the stress "RX ping on STA, Start-stop uAP" in loop.
  - Added 11AX macro so that nss settings can be available for 11AC as well as 11AX based configurations.
  - Fixed various warning for IAR, MDK, armgcc and mcuXpresso compilers.

Version 1.3.r43.p5

- Updates:
  - Updated firmware version to p168(however signed firmware is not updated in this release and has version p164).
- Bug Fixes:
  - Fixed: DUT not able to connect to AP configured in WPA2/WPA3 mixed mode security.
  - Fixed: Association Request does not include Group Management Cipher Suite IE in RSNIE when PMF mandatory is set on STA in WPA2-PSK/WPA3/OWE security.
  - Fixed: uAP Beacons does not include SHA256 in RSNIE when PMF mandatory is set on AP in WPA2-PSK security.
  - Fixed: Device getting hanged while setting txpwrlimit by CMD 'wlan-set-txpwrlimit'.
  - Fixed: Not able to load RU tx power limit with RT.
  - Fixed: Not able to change HE parameters in HE MAC Capabilities or HE PHY Capabilities using 11axcfg command.

Version 1.3.r43.p4

- Updates:
  - None.
- Bug Fixes:
  - Updated helper message for owe_only cli.
  - Fixed: Observed bi-directional data traffic on air while running uni-directional TCP-RX traffic.
  - Fixed: UDP-TX traffic going without using aggregation.
  - Throughput values will now be displayed on DUT while running iperf Tx and Rx traffic.

Version 1.3.r43.p3

- Updates:
  - Updated firmware version to p164.
- Bug Fixes:
  - Added ed mac config support for uAP.
  - Modified process_rsn_ie function on driver side to provide support for Group Management Cipher suite.
  - Added conditional logic related to handling of region code.

Version 1.3.r43.p2

- Bug Fixes:
  - Added delay in uart_wifi_bridge application to correctly read calibration data for RB3P board.

Version 1.3.r43.p1

- Updates:
  - Updated firmware version to p162.
  - Added MBO feature support.
  - Added OWE feature support.
- Bug Fixes:
  - uAP Beacons contains MFPC & MFPR bits set to 1 in WPA2-PSK security after removing the previous WPA3 security profile.

  - Added new cli parameters for wlan-set-tx-omi command to send OMI using QoS Null Packet or QoS Data Packet according to input provided.
  - Corrected 11R configuration macro in allMacros_iw61x.txt file and removed the same from ignoreMacros_iw61x.txt file.
  - Coverity fix: Changed all WM_FAIL to -WM_FAIL.

Version 1.3.r42.p4 -New Additions:

- CA2 and RB3+ f/w updated to v16_xx.21.p64.1

Version 1.3.r42.p3 -New Additions:

- 8801 f/w update -Bug Fixes:
- Fixed Misra issues.

Version 1.3.r42.p2 -Bug Fixes:

- Fixed Misra issues.

Version 1.3.r42.p1 -New Additions:

- RTOS abstraction improvement
- Added support for FIPS for CA2 and RB3+
- Added uAP 11AC support
- Deprecated support for 88W8977 Wi-Fi SoC
- Added 8978 SoC firmware p64.
- Added 8987 SoC firmware p64.
- Added mlanutl equivalent utility on RT to create wifi fw compatible command arrays.

-Bug Fixes:

- Fixed Misra issues.

Version 1.3.r41.p2 -New Additions:

- Updated license header as a github friendly license.
- Added 8801 SoC firmware p177 with get Coex Statistics support.
- Added 8987 SoC firmware p32.2 with Bluetooth related fixes. Added fix for automatic ble disconnect issue.

-Bug Fixes:

- Fixed Unable to start/stop 11n rx reorder timer(50%).
- Fixed Getting Improper Channel Number in "wlan-get-uap-channel" command.
- Fixed DUT fails to start DHCP intermittently.
- Fixed wrong wlan-list output.
- Fixed station connect issue after host sleep is enabled and disabled.
- Fixed Misra and coverity issues.
- Fixed automatic ble disconnect issue after 30 seconds of connection establishment.

Version 1.3.r41.p1 -Bug Fixes:

- Updated 8801 SoC firmware to toggle GRANT Pin with WLAN/BT time on HIGH Request with Low Priority.

- Added coverity fixes.

Version 1.3.r40.p5 -Bug Fixes:

- Fixed issue of DUT not entering Powersave mode on all SoCs.

Version 1.3.r40.p4 -Bug Fixes:

- Default bandwidth set to 20 MHz for uAP on 8977.
- Fixed an issue in static IPv4 address assignment.
- Fixed warning and errors for IAR, MDK and gn + ninja + armgcc toolchains.

Version 1.3.r40.p3 -Bug Fixes:

- Added wlan_uap_set_httxcfg API to set 40 MHz support in 2.4 GHz.
- Fix for build issue seen for wifi_setup 1020

Version 1.3.r40.p2 -Bug Fixes:

- Fixed IPv6 address and state updates as per the networking stack configuration.
- Removed errors shown in channel validation and in setting custom CFP tables
- Added API to support selection of 20 and 40 MHz bandwidth
- Added -b(bandwidth) option in wifi_cli iperf command
- DHCP client doesn't report the failure of obtaining IP address
- Remove dependency for wifi_config_internal.h
- Remove warnings reported for channel list variables
- Fix STA not sending data in 40M bandwidth for BGN 40 in 2.4G

Version 1.3.r38.p2

- New Additions:
  - Added Support for IPv6.
  - Added support for Hostsleep and packet filters.
  - Updated Firecrest Firmware version to p50.5.
  - Updated CA2 Firmware version to p21.22.

-Bug Fixes:

- Fixed an issue where DUT not able to start UAP Network in 2G MMH Mode.
- Fixed an issue where Command 0xb2 timeout is seen when stopped uAP while running DL traffic.

Version 1.3.r37.p4

- New Additions:
  - Added support for new API for HostCmd in RT Platform.
  - Added provision to set tx rates for HE mode in wlan_set_txratecfg API.
  - Added support for new command to set tx OMI.
  - Added WPA3 R3 support for 8801, RB3, RB3+ and CA2.
  - Updated Firecrest firmware version to p50.2.
  - Updated CA2 firmware version to p235.2
  - Updated RB3+ firmware version to p11.3.
  - Updated 8801 firmware version to p191.2
  - Updated RB3 firmware version to p186.2.

- Bug Fixes:
  - Fixed an issue where default netif was not set to STA after closing uAP.
  - Fixed an issue where 30% low RX throughput was seen with 11AN 40Mhz.
  - Fixed wlan_start and wlan_stop API working.
  - Fixed High ping latency when DUT is put in IEEEPS mode.
  - Fixed an issue where wlan_get_dtim_period API was not returning any value.
  - Fixed SVD vulnerability issue on RB3+.
  - Fixed MISRA/Coverity issues.
  - Fixed SVD vulnerability issue on 8977, 8801 and CA2.
  - Fixed an issue in Firecrest where STAUT is disconnecting immediately after ieeeps command is fired.
  - Fixed an issue where UDP traffic was not working on uAP mode.
  - Country code not being displayed in 11d is being fixed

Version 1.3.r35.p2

- New Additions:
  - Added support for 11ax for Firecrest.
  - Updated Firecrest firmware version to p27.
- Bug Fixes:
  - Fixed an issue for CA2 where tcp/udp Rx traffic was not seen with Linksys AP.
  - Fixed tx power limit issue for Firecrest.
  - Fixed an issue for Firecrest where Ping stops working after TCP traffic is started from DUT.
  - Changed WLAN_PSK_MAX_LENGTH to 64 from 65.
  - Fixed an issue where mfpc was not set properly.

Version 1.3.r34.p2

- New Additions:
  - Updated CA2(8987) SoC firmware to p235.1.
- Bug Fixes:
  - Fixed WPA3 Authentication failure for CA2.
  - Fixed SPP connection issue.

version 1.3.r34.p1

- New Additions:
  - Updated CA2(8987) SoC firmware to p235.
  - Added Separation of fw download from Wi-Fi initialization.
  - Added support for Multicast group creation
  - Updated license content in the wlan src for Murata or Generic customer.
  - Added support for new firmware version display.
  - Added config Macro for disabling Rx SDIO aggregation. This is enabled by default.
- Bug Fixes:
  - Fixed CH 144 connection issue.

version 1.3.r33.p2

- New Additions:

  - Updated RB3+(8978) SoC firmware to p214.
  - Updated CA2(8987) SoC firmware to p200.
- Bug Fixes:
- Fixed an issue where, wifi connection was failing during BLE activity.
- Fixed an issue where a2dp profile app could not find a2dp sink device.

version 1.3.r32.p5

- New Additions:
  - Updated RB3+(8978) SoC firmware to p198.
  - Updated CA2(8987) SoC firmware to p199.
- Bug Fixes:
  - Fixed an issue where BT pairing auth failure was observed between two 8987 or two IW416 devices due to same DH Public Key being generated.

version 1.3.r32.p4

- New Additions:
  - Updated RB3+(8978) SoC firmware to p197.
- Bug Fixes:
  - Fixed an issue where, in presence of WLAN, BT A2DP SNK scenario can observe continuous glitches because of BT utilizing the WLAN overlapping channels.
  - Fixed an issue where, in presence of BT A2DP SRC scenario and WLAN traffic, both will share air-time. WLAN Rx TP can drop to 15-20% of its baseline because of A2DP occupying more air-time duty cycle.

version 1.3.r32.p3

- New Additions:
  - Updated RB3+(8978) SoC firmware to p196.
  - Updated CA2(8987) SoC firmware to p162.
  - Updated RB3(8977) SoC firmware to p186.
  - Updated Falcon(8801) SoC firmware to p191.
- Bug Fixes:
  - Fixed an issue where STAUT was not advertising extended capabilities in assoc request
  - Fixed an issue where cal data download API is not setting the cal data correctly.
  - Fixed an issue where Ping of 10000 bytes is not workingfor WMM case with 11n/11ac pre-cert WFA testbed AP's.
  - VU FFD(vulnerability) fixes have been made for CA2(8987)/RB3+(8978)/RB3(8977)/-Falcon(8801).

version 1.3.r31.p1

- New Additions:
  - Updated RB3+(8978) SoC firmware to p152.
  - Updated CA2(8987) SoC firmware to p152.
  - Added WMM feature for 8987 SoC -Updates
  - Enabled wifi deep sleep and IEEEPS modes as part of CLI initialisation.
  - Updated module macro for CA2 from WIFI_BOARD_AW_CM358MA to WIFI_BOARD_-

AW_CM358.
- Bug Fixes:
  - Fixed deepsleep error when called immediately after disconnection
  - Fixed an issue where uAP was not turned on when country is specified using wlan_set_country API.

version 1.3.r30.p2

- New Additions:
  - Updated 8978 SoC firmware to p185.
  - Updated 8987 SoC firmware to p185.
- Bug Fixes:
  - Fixed a regression issue in WiFi FW where Ex-client is not able to associate with uAP on 8978 and 8987 SoC.

version 1.3.r30.p1

- New Additions:
  - Updated RB3+(8978) SoC firmware to p145.
  - Updated CA2(8987) SoC firmware to p145.
- Bug Fixes:
  - Fixed MISRA C-2012 Rule 14.4 issues.
  - Updated TX Power configuration table for uAP.
  - Fixed an issue where in MFG mode measured TX power value for 2.4GHz is 10.97dBm and for 5GHz is 12.03dBm always irrespective of TX-power values configured in CA2.
  - Fixed an issue where in control frames measured TX power is less than configured Tx power by 3dBm for 5Ghz 40Mhz BW and ∼7dBm for 5Ghz 80Mhz BW in CA2.

version 1.3.r29.p2

- New Additions:
  - Updated 8987 SoC firmware to p142.
  - Updated 8978 SoC firmware to p142.
- Bug Fixes:
  - Fixed MISRA C-2012 Directive 4.7, Directive 4.10, Rule 15.7 and Rule 14.4 issues
  - Fixed wlan_get_sta_tx_power() API implementation for retrieving station tx power level.
  - Fixed wlan-set/get-txpwrlimit CLI command for setting/getting TX power limit for 11AC modulation groups in 8987.
  - Fixed an issue where in MFG mode(rf_test_mode=1) after disabling wlan_set_rf_cont_-mode(CMD18_CW=1) command response timeout is seen in 8987.

version 1.3.r29.p1

- New Additions:
  - Added support for 8987 SoC with p141 firmware version.
  - Updated 8978 SoC firmware to p141.
- Bug Fixes:
  - Fixed Misra C-2012 required category issues for Rule 17.7.
  - Fixes for BCA-TDM in Co-Ex for 8978 SoC.

  - Fixed A2DP glitches in BT when WLAN connected for 8978 SoC.

version 1.3.r27.p2

- New Additions:
  - Updated FW versions to p130 for 8978.

version 1.3.r27.p1

- Bug Fixes:
  - Fixed PMF pre-cert issue where STAUT is not associating to PSK-SHA-256 enabled AP.
  - Fixed 11N pre-cert issue where 11N-5.2.47 STAUT AMPDU TX test case is failing.

version 1.3.r26.p2

- New Additions:
  - Updated FW versions to p184 for 8977 and p122 FW for 8978.
  - Added wifi_cert application under wifi_examples.
- Bug Fixes:
  - Fixed an issue where connection problem is seen with uAP in wifi_webconfig after removing stored credentials.
  - Fixed RF Test Mode issue for setting data rate in uAP mode.
  - Fixed Coverity and MISRA issues in WiFi Driver.
  - Fixed WPA3 SAE pre-cert requirement where there was requirement of Auth confirm to be initiated by either STA or Ex-AP.
  - Removed following API's from WiFi driver as they were not supported:
    * wifi_auto_reconnect_enable()
    * wifi_auto_reconnect_disable()
    * wifi_get_auto_reconnect_config()
    * wifi_get_tbtt_offset()
    * wifi_set_packet_filters()
    * wifi_set_auto_arp()
    * wifi_tcp_keep_alive()
    * wifi_nat_keep_alive()

version 1.3.r23.p2

- New Additions:
  - Minor update to wifi_test_mode CLIs for better usability.
  - Added bug fixes in WiFi FW and updated FW versions to p155 for 8801, p182 for 8977 and p106 for 8978.
  - Added new wlan_uap_set_htcapinfo() API for setting HT Capability field for uAP.
- Bug Fixes:
  - Fixed RF Test Mode issues for SD8801 and SD8977 reported by QA.
  - Fixed WiFi 802.11n WPA3 SAE pre-cert test failures for SD8978.
  - Fixed stack overflow issue with WLCMGR thread during wlan-connect.
  - Fixed memory corruption issue cause by scan list overflow when using 11D.

version 1.3.r21.p1

- New Additions:

  - Added support for SD8978.
  - Added Test Mode support for 8801, 8977 and 8978.
  - Added new FW binaries for 8801, 8977 and 8978.
  - Added OTP Force Region support in WiFi Driver.
  - Added support for DHCP Server CLI to print IP addresses of connected clients to uAP.
  - Added support to set HT Capability field for uAP.
  - Added wlan_get_chanlist API and CLI.
  - Added WiFi Driver task priority configurability option.
  - Reduced WiFi Driver SRAM footprint.
  - Added support in Wi-Fi driver to print debug events from WLAN FW.
  - Added support for FW Dump generation using a micro-USB mass storage device.
- Bug Fixes:
  - Fixed an issue where a redundant Link Loss disconnect timer was defined in the wifi_iperf app.
  - Fixed an issue where sometimes the wlan_disconnect call did not abort an ongoing re-association.
  - Fixed an issue where station connection to an Open security Ext-AP fails after connection to a WPA3-SAE Ext-AP.
  - Fixed an issue where uAP did not start with WPA3-SAE security type.
- Known Issues
  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r20.p1

- New Additions:
  - Added new FW for 8801.
- Bug Fixes:
  - Fixed an issue where STAUT went to hang state when doing a disconnect in the wifi_iperf app.
  - Fixed an issue where STAUT failed to reassociate with an Ext-AP after band switch.
  - Fixed an issue where Scan command timeout was observed after changing the ssid of Ext-AP to which STAUT is connected.
- Known Issues
  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r19.p1

- New Additions:
  - Added new FW for 8801.
  - Updated WiFi API reference manual.
- Bug Fixes:
  - Fixed an issue where STAUT attempted to connect with Ext-AP continuously even when an Auth Failure occured due to incorect passphrase.
  - Fixed an issue where SDIO write error occured sometimes for some packet sizes during Tx.
  - Fixed an issue where packet leakage was noticed on SD8801 when noise was applied.
  - Fixed an issue to supress Association Failed warning generated during a connection attempt.
- Known Issues

  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r18.p1

- New Additions:
  - Added new FW for 8977.
- Bug Fixes:
  - Fixed an issue where STAUT fails to scan after Channel Switch and STAUT moves to new channel after eCSA IE.
  - Fixed IAR, MDK build compilation warnings.
- Known Issues
  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r17.p1

- New Additions:
  - Added new FW for 8977.
  - Added CLI command to get a list of stations connected to uAP.
- Bug Fixes:
  - Fixed an issue where Scan was temporarily aborted due to noise but correct status was not being returned to the host.
  - Fixed an issue where STAUT failed to connect with an AP with hidden SSID.
  - Fixed IAR, MDK build compilation warnings.
  - Fixed an issue where TCP bi-directional throughput stops abruptly for SD8801 and STAUT is disconnected.
  - Fixed an issue where UDP Dual Mode throughput stops abruptly and disconnection is seen.
  - Fixed an issue where Tx got stuck after removal of interference noise.
- Known Issues
  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r16.p1

- New Additions:
  - Added PMF Configuration for uAP.
  - Updated WLAN Versioning.
  - Updated WLAN CLIs.
- Bug Fixes:
  - Fixed an issue where the STAUT sometimes did not send aggregated packets during Tx.
  - Fixed an issue External Client could not associate with the UAP in WPA3-SAE mode.
  - Fixed IAR build compilation warnings.
  - Fixed an issue where the STAUT failed to associate with an AP with hidden SSID.
  - Fixed an issue where active scan probe requests were sent for DFS channels.
  - Fixed an issue where 2.4GHz-HT40 power values were exposed via CLI configuration.
  - Fixed an issue where the STAUT will be continuously in re-association mode after link-loss.
- Known Issues
  - Wi-Fi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r15.p1

- New Additions
  - Added support for Tx Power Limit configuration.
  - Added support for Channel List configuration.
  - Added support for CW MODE.
  - Added support for sysinfo CLI to get threads information, network stats, wlan stats and heap stats.
  - Added -d and -r options to iperf CLI for dual and trade-off mode.
  - Added support for antenna configuration for 8801.
  - Added support for band configuration.
  - Added new FW for 8977
- Bug Fixes:
  - Fixed an issue where UDP Rx data rate was low in iperf dual mode operation.
  - Fixed an issue where STAUT traffic was getting halted when there is traffic in overlapping BSS on the extension channel.
  - Fixed an issue where STAUT was not able to transmit above MCS 4.
  - Fixed an issue where association with uAP failed with WPA2 security mode.
  - Fixed an issue where STAUT failed to connect to WPA3 AP due to PMF config mismatch.
  - Fixed an issue where ping loss was observed for packets of size greater than 10000 bytes.
  - Fixed an issue in CLI where CR and LF characters where not handled properly.
  - Fixed an issue where TCP-Tx traffic abruptly stops when parallel traffic is ongoing on another device using the same AP.
  - Fixed an issue where DUT goes into hang state when iperf run is aborted.
  - Fixed an issue where the STATU was not sending aggregated packets to the AP.
  - Fixed an issue where UAP did not start with WPA2 security mode when ACS is configured.
  - Fixed an issue where ED MAC was not enabled by default.
- Known Issues
  - WiFi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r14.p1

- New Additions
  - Added support for Panasonic PAN9026 module.
  - Added -t option in iperf CLI for setting traffic running time.
  - Added -B option for supporting Tx/Rx Multicast packets during iperf runs.
  - Added World Wide Safe Mode configurability to the WiFi Driver.
  - Added ED MAC support for 8977.
  - Added support for PMF APIs and CLIs.
  - Added new FW for 8977 and 8801.
- Bug Fixes:
  - Fixed Coverity and mandatory MISRA issues reported on v1.3.r13.p1 release.
  - Fixed an isuue where DUT console was getting stuck after intiating connection with an AP which has different RSN values than expected.
  - Fixed an issue where DUT was not able to roam from SAE-PSK to PSK configured AP.
  - Fixed an issue where the app became unresponsive after Soft AP is stopped.
  - Removed unnecessary files after Blackduck scan.
- Known Issues

  - WiFi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r1r3.p1

- New Additions
  - Updated WiFi Driver to be independednt of the LwIP stack. Only the WLAN Connection Manager now uses LwIP.
  - Added WiFi Roaming feature.
  - Added CLI support for iperf in wifi_cli app.
  - Added support for CSA handling from wlan station side.
  - Added WLAN APIs for wlan-set-txratecfg, wlan-get-txratecfg, wlan-get-data-rate, wlan-set-reg and wlan-get-reg.
  - Updated External AP SSID required for wifi_iperf app.
- Bug Fixes:
  - Fixed Coverity issues reported on v1.3.r12.p1 release.
  - Added a fix for ping loss observed during ping test.
  - Added a fix where the console became unresponsive after wlan-start-network+wlan-stop-network commands are executed in loop.
  - Added a fix for HT IE missing from beacon for both 2.4GHz and 5GHZ bands.
  - Fixed warnings reported for IAR.
  - Added a fix for increasing DHCP leave time to 24 hrs for long duration stress tests.
- Known Issues
  - WiFi sample apps do not work with Rev-C (2018) version of RT685 board

version 1.3.r12.p1

- New Additions
  - Added support for ping utility.
- Bug Fixes
  - Restart of SoftAP fails once SoftAP is stopped.
  - HT IE is missing from beacon for both 2.4GHz and 5GHZ bands.
  - Low TCP-Rx and UDP-Rx throughput observed for 5GHz-HT40 band.
  - uAP couldn't be started after STA is associated to Ex-AP.
  - Stopping SoftAP results in disconnection of In-STA from Ex-AP.
  - DHCP server is not started for BSS created using "wlan-add" command.
- Known Issues
  - N.A

version 1.3.r11.p2

- New Additions
  - Updated SDK version for RT1060 to 2.7.1 RFP RC2.
  - Added FP91 based WiFi FW for SD8801, SD8977, SD8978, and SD8987.
  - SD8801, SD8977, SD8978 and SD8987 WiFi Fw have embedded WPA3 SAE support from version v1.3.r11.p1 onwards.
  - Added WLAN CLI support. Added new wifi_cli for CLI demo.
- Enhancements
  - License and Copyright Updates.

  - Formated code base with clang-format 10.0.0 standards.
  - Integrated PICK utility for WiFi Driver code scrubbing.
- Bug Fixes
  - Removed unnecessary prints LwIP stats display.
  - Fixed an issue where uAP did not start on 5GHz band.
  - Fixed an issue where loww TCP/UDP-Rx throughput was observed for 5GHz-HT40Mhz.
- Known Issues
  - N.A

version 1.3.r10.p1

- New Additions
  - Moved to using LwIP provided IPerf App.
  - Added SDIO Rx Aggregation support to improve throughput.
  - Added support for 11ac configuration API.
  - Updated License and Copyright information on all WiFi driver files and sample apps.
  - Moved MCUXpresso SDK base for RT1060 platform from 2.6.2 to 2.7.0 rc3.
- Known Issues
  - N.A

version 1.3.r9.p1

- New Additions
  - Achieved additional TP improvements by updating LwIP parameters.
- Bug Fixes
  - Fixed the issue of low throughput for both TCP and UDP (Tx/Rx) by updating LwIP parameters.
  - TCP Throughput was observed to be lower than UDP throughput by 70%-80%. Fixed this issue by updating TCP related LwIP configurations.
- Known Issues
  - N.A

EdgeFast Bluetooth Protocol Abstraction Layer Implementation for MCUXpresso SDK

The current version is 1.6.0.

- 1.6.0
  - Improvement:
    - Update source files to align Zephyr V3.6 base.
    - The name of edegfast_bluetooth configuration files are changed from app_config.h to app_bluetooth_config.h.
  - New features:
    - Add a2dp_bridge function on RT1170-EVKB UMS demo.
    - Add new shell commands for UMS/UMR, BMS/BMR to improve test efficiency and user experience.
    - Support IW612(Murata Type 2EL-M2) on RT595-EVK, RT1050-EVK and RT685-E-

VK boards.
    - Enable a2dp_sink, a2dp_source, handsfree and handsfree_ag on RT1050-EVK.
    - Low power feature supported on RT595+1XK/2EL in peripheral_ht/central_ht examples.
    - Independent reset feature supported on RT1170-EVKB and RT1060-EVKC.
- 1.5.1
  - New features:
    - LE audio support on RT1060-EVKC.
- 1.5.0
  - Improvement:
    - Update source files to align Zephyr V3.5 base.
- 1.4.0
  - New features:
    - Add example wifi_cli_over_ble_wu
    - Add following LE audio examples, including,
      - wifi_cli_over_ble_wu
      - broadcast_media_sender and broadcast_media_receiver
      - unicast_media_sender and unicast_media_receiver
      - tmap_central and tmap_peripheral
      - call_gateway and call_terminal
    - Update the shell example to support LE audio features.
- 1.3.1
  - Improvement:
    - Fix MISRA and Coverity issue.
    - Fix building warnings.
    - Fix autopts issues.
    - Improve audio_profile aws part based on new middleware/aws_iot.
  - New features:
    - Support MCUXpresso IDE NPW
    - Support OpenCMSIS Pack
- 1.3.0
  - Improvement:
    - Updated to Zephyr Bluetooth 3.2.0.
    - Fix autopts issues.
    - Improve audio_profile aws part based on new middleware/aws_iot/iot-reference/examples/evkbmim example.
  - New features:
    - Enable EVKMIMXRT1160
    - Default Bluetooth module is changed to Murata
  - Others:
    - Remove WIFI_provisioning example.
- 1.2.1
  - Improvement:
    - Fix autopts issues.
    - Support creating audio streaming without calling in handsfree_ag.

  - New features:
    * Enable EVKMIMXRT1040.
    * Support muRata modules on EVKMIMXRT1040, EVKBIMXRT1050 and EVKBIMXRT1060.
- 1.2.0
  - Improvement:
    * Updated to Zephyr Bluetooth 3.0.0
    * Fix autopts issues.
  - New features:
    * Enable EVKBIMXRT1050
    * Enable peripheral_beacon on all boards.
- 1.1.1
  - New features:
    * Enable peripheral_beacon example in EVKMIMXRT1170 board.
- 1.1.0
  - Improvement:
    * Updated to Zephyr Bluetooth 2.6.0
    * change fixed wireless_uart central and peripheral role number to flexiable number.
- 1.0.0
  - New features:
    * Enable SMP Pairing and Bonding feature for BLE
    * Enable AVRCP
    * Enable L2CAP for BR
    * Add examples wireless_uart and edgefast_bluetooth_shell
    * Enable data signing, GATT caching, GATT service changed features
    * Enable data len update, whitelist, phy update
    * Enable privacy mode
    * Enable setting for bonding
    * Enable A2DP content protection/recovery/reporting/delay reporting/header compression/multiplexi services.
  - Improvement:
    * Enhance SPP to support multiple connection
    * Use LittleFS to manage the NVM.
    * Improve A2DP APIs and implementation.
      · Add endpoint callbacks to struct bt_a2dp_endpoint.
      · Add content protection/recovery/reporting/delay reporting/header compression/multiplexing related fields to struct bt_a2dp_endpoint and struct bt_a2dp_control_cb.
      · Upgrade struct a2dp_configure_result to struct bt_a2dp_endpoint_configure_result.
      · Add deconfigured callback.
      · Change sink_start_play callback to start_play callback.
      · Change sink_suspend_play callback to stop_play callback.
      · Add bt_a2dp_disconnect API.
      · Remove bt_a2dp_register_control_callback API.
      · Change bt_a2dp_configure API's callback.

  - Add peer_endpoint to bt_a2dp_configure_endpoint API.
  - Change bt_a2dp_src_start to bt_a2dp_start.
  - Change bt_a2dp_src_suspend to bt_a2dp_stop.
  - Remove bt_a2dp_get_configured_peer_endpoint.
  - Change the parameter as endpoint for the APIs: bt_a2dp_deconfigure, bt_a2dp_start, bt_a2dp_stop, bt_a2dp_reconfigure, bt_a2dp_src_media_write and bt_a2dp_snk_media_sync.
  - Add new APIs for content protection/delay reporting services: bt_a2dp_set_cp_header, bt_a2dp_set_initial_delay_report and bt_a2dp_send_delay_report.

- 0.1.0
  - New features:
    - Initialization version

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