MCUXpresso SDK Release Notes Supporting valmimxrt600

Change Logs

NXP Semiconductors

Contents

Driver Change Log

Title

Page No.

Middleware Change Log

Component Change Log

Title

Page No.

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.3.0.

- 2.3.0
  - Added CLOCK_GetFreq() API
  - Removed DTRNG flag wait in clock API to avoid unconditional delay. DTRNG will be reloaded in crypto init function.
- 2.2.0
  - Added els_gdet clock source enumeration
  - Fixed kMAIN_CLK_to_DMIC_CLK value
- 2.1.4
  - Added noinline attribute to CLOCK_Delay() to work around compiler optimization issue
- 2.1.3
  - Added delay for DTRNG busy flag before disabling T3 256M
- 2.1.2
  - Renamed kCLOCK_Css/kCLOCK_CssApb to kCLOCK_Els/kCLOCK_ElsApb
- 2.1.1
  - Added ANA_GRP XTL power control in USB PHY API
- 2.1.0
  - Added USIM_CLOCKS macro
  - Added CLOCK_DisableUsbhsPhyClock API
- 2.0.1
  - Moved g_clkinFreq and g_mclkinFreq inside extern "C"
- 2.0.0
  - initial version.

POWER

The current POWER driver version is 2.5.0.

- 2.5.0
  - Added Power_InitLoadGdetCfg() API
  - Added bool return value for POWER_EnableGDetVSensors()
- 2.4.0
  - Added POWER_TrimSvc() API
  - Added pack parameter to POWER_InitVoltage() API
  - Moved POWER_DelayUs() to execute in SRAM
  - Added barriar around WFI
  - Tweaked SVC table
  - Fixed POWER_GetResetCause() to get correct cause
- 2.3.0

  - Added POWER_SetPowerSwitchCallback() API
  - Added POWER_InitVoltage() API
  - Remove PMIP_BUCK_LVL configuration from POWER_InitPowerConfig().
  - Fixed a potential compiling issue in Power_Delay()
- 2.2.0
  - Added POWER_DisableGDetVSensors() API
  - Added POWER_EnableGDetVSensors() API
  - Added GDET/VSensor setting around PM2 PM3
- 2.1.1
  - Renamed kPOWER_Pm2MemPuCss to kPOWER_Pm2MemPuEls
  - Supported PM3 wakeup on A1 device
- 2.1.0
  - Added PM3 wakeup support for A1 device.
  - Added POWER_ConfigCauInSleep() API. Remove pm3CauPd field from power_sleep_config_t structure.
  - Added power_init_config_t parameter in POWER_InitPowerConfig() API.
- 2.0.1
  - Improved power performance
  - Added return value for POWER_EnterPowerMode()
- 2.0.0
  - initial version.

IO_MUX

The current IO_MUX driver version is 2.2.0.

- 2.2.0
  - Update io_mux signals according to data sheet.
- 2.1.2
  - Fixed misra issues
- 2.1.1
  - Added driver strength configuration
- 2.1.0
  - Added IO_MUX_SetPinOutLevelInSleep API
  - Added IO_MUX_SetRfPinOutLevelInSleep API
  - Added capture pulse macro IO_MUX_AON_CAPTURE
- 2.0.0
  - initial version.

RESET

The current RESET driver version is 2.1.1.

- 2.1.1

  - Corrected XX_RSTS definitions.
  - Removed USDHC_RSTS.
- 2.1.0
  - Added RESET_ReleasePeripheralReset() API
- 2.0.3
  - Renamed CSS(Crypto subsystem) related macros to ELS(Edge lock security)
- 2.0.2
  - Added USIM_RSTS macro
- 2.0.1
  - Added kCSS_GDET_REF_RST_SHIFT_RSTn
- 2.0.0
  - initial version.

OCOTP

The current OCOTP driver version is 2.2.1.

- 2.2.1
  - Removed reset on OTP init and deinit to keep OTP configuration on boot.
- 2.2.0
  - Added OCOTP_ReadPackage() API
  - Exposed OCOTP_ReadSocOtp() API
- 2.1.0
  - Added OCOTP_ReadSVC() API.
  - Avoid access OTP register before busy wait in OCOTP_OtpFuseRead()
- 2.0.1
  - Fixed an misra issue 10.1
- 2.0.0
  - initial version.

ROMAPI

The current ROMAPI driver version is 2.0.0.

- 2.0.0
  - initial version for A0.

IPED

The current IPED driver version is 1.0.0.

- 1.0.0
  - initial version

I2S_BRIDGE

The current I2S_BRIDGE driver version is 2.0.0.

- 2.0.0
  - initial version

CNS_ACOMP

The current driver version is 2.0.1.

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

CNS_ADC

The current driver version is 2.2.0.

- 2.2.0
  - Improvements
    - Updated cns adc trigger sources.
    - Migrated cns adc trigger sources enumeration from cns_adc.h to device.h
    - Reserved single-end mode channel 15, differential mode channel 5, and channel 15.
- 2.1.0
  - Bug Fixes
    - Fixed temperature measurement error, and provided 'enableChop' member to control the ADC chop.
- 2.0.2
  - Bug Fixes
    - Fixed ADC scan channel misconfiguration issue.
    - Fixed violation of MISRA C-2012 rule 10.1 and rule 10.4.
  - Improvements
    - Added new member "enableInputBufferChop" into "adc_config_t" to enable/disable input buffer chopper.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA-2012 rules.
      - Rule 14.2, rule 10.3.
- 2.0.0
  - Initial version.

CACHE

The current CACHE driver version is 2.0.8.

- 2.0.8
  - Improvements
    - Updated function CACHE64_GetInstanceByAddr() to support some devices that provide alias of cacheable memory section.
- 2.0.7
  - Improvements
    - Check input parameter "size_byte" must be larger than 0.
- 2.0.6
  - Bug Fixes
    - Fixed overflow for CACHE64_GetInstanceByAddr()/CACHE64_CleanCacheByRange()/CACHE64_InvalidateCacheByRange() APIs.
- 2.0.5
  - Improvement
    - Made use of FSL_FEATURE_CACHE64_CTRL_HAS_NO_WRITE_BUF feature
- 2.0.4
  - Improvement
    - Disable cache policy feature on SoC without CACHE64_POLSEL IP.
  - Bug Fixes
    - Fixed doxygen issue.
- 2.0.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.3.
- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3, 10.4 and 14.4.
    - Fixed doxygen issue.
- 2.0.1
  - Improvements
    - Moved CLCR register configuration out of the while loop, it's unnecessary to repeat this operation.
- 2.0.0
  - Initial version.

COMMON

The current COMMON driver version is 2.4.2.

- 2.4.2
  - Improvements
    - Add the macros to convert peripheral address to secure address or non-secure address.
- 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.

CRC

The current CRC driver version is 2.1.1.

- 2.1.1
  - Fix MISRA issue.
- 2.1.0
  - Add CRC_WriteSeed function.
- 2.0.2
  - Fix MISRA issue.
- 2.0.1
  - Fixed KPSDK-13362. MDK compiler issue when writing to WR_DATA with -O3 optimize for time.
- 2.0.0
  - Initial version.

CTIMER

The current CTimer driver version is 2.3.1.

- 2.3.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.7 and 12.2.
- 2.3.0
  - Improvements
    - Added the CTIMER_SetPrescale(), CTIMER_GetCaptureValue(), CTIMER_EnableResetMatchChannel(), CTIMER_EnableStopMatchChannel(), CTIMER_EnableRisingEdgeCapture(), CTIMER_EnableFallingEdgeCapture(), CTIMER_SetShadowValue(),APIs Interface to reduce code complexity.
- 2.2.2
  - Bug Fixes
    - Fixed SetupPwm() API only can use match 3 as period channel issue.
- 2.2.1
  - Bug Fixes
    - Fixed use specified channel to setting the PWM period in SetupPwmPeriod() API.
    - Fixed Coverity Out-of-bounds issue.
- 2.2.0
  - Improvements
    - Updated three API Interface to support Users to flexibly configure the PWM period and PWM output.
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 8.4.
- 2.1.0
  - Improvements
    - Added the CTIMER_GetOutputMatchStatus() API Interface.

        - Added feature macro for FSL_FEATURE_CTIMER_HAS_NO_CCR_CAP2 and FSL_FEATURE_CTIMER_HAS_NO_IR_CR2INT.
- 2.0.3
    - Bug Fixes
        - MISRA C-2012 issue fixed: rule 10.3, 10.4, 10.6, 10.7 and 11.9.
- 2.0.2
    - New Features
        - Added new API "CTIMER_GetTimerCountValue" to get the current timer count value.
        - Added a control macro to enable/disable the RESET and CLOCK code in current driver.
        - Added a new feature macro to update the API of CTimer driver for lpc8n04.
- 2.0.1
    - Improvements
        - API Interface Change
            - Changed API interface by adding CTIMER_SetupPwmPeriod API and CTIMER_UpdatePwmPulsePeriod API, which both can set up the right PWM with high resolution.
- 2.0.0
    - Initial version.

CNS_DAC

The current driver version is 2.1.0.

- 2.1.0
    - Improvements
        - Updated cns dac trigger sources.
        - Migrated cns dac trigger sources enumeration from cns_dac.h to device.h
- 2.0.1
    - Bug Fixes
        - Fixed violations of the MISRA C-2012 rules.
- 2.0.0
    - Initial version.

DMA

The current DMA driver version is 2.5.3.

- 2.5.3
    - Improvements
        - Add assert in DMA_SetChannelXferConfig to prevent XFERCOUNT value overflow.
- 2.5.2
    - Bug Fixes
        - Use separate "SET" and "CLR" registers to modify shared registers for all channels, in case of thread-safe issue.

- 2.5.1
  - Bug Fixes
    - Fixed violation of the MISRA C-2012 rule 11.6.
- 2.5.0
  - Improvements
    - Added a new api DMA_SetChannelXferConfig to set DMA xfer config.
- 2.4.4
  - Bug Fixes
    - Fixed the issue that DMA_IRQHandle might generate redundant callbacks.
    - Fixed the issue that DMA driver cannot support channel bigger then 32.
    - Fixed violation of the MISRA C-2012 rule 13.5.
- 2.4.3
  - Improvements
    - Added features FSL_FEATURE_DMA_DESCRIPTOR_ALIGN_SIZEn/FSL_FEATURE_DMA0_DESCRIPTOR_ALIGN_SIZE/FSL_FEATURE_DMA1_DESCRIPTOR_-ALIGN_SIZE to support the descriptor align size not constant in the two instances.
- 2.4.2
  - Bug Fixes
    - Fixed violation of the MISRA C-2012 rule 8.4.
- 2.4.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 5.7, 8.3.
- 2.4.0
  - Improvements
    - Added new APIs DMA_LoadChannelDescriptor/DMA_ChannelIsBusy to support polling transfer case.
  - Bug Fixes
    - Added address alignment check for descriptor source and destination address.
    - Added DMA_ALLOCATE_DATA_TRANSFER_BUFFER for application buffer allocation.
    - Fixed the sign-compare warning.
    - Fixed violations of the MISRA C-2012 rules 18.1, 10.4, 11.6, 10.7, 14.4, 16.3, 20.7, 10.8, 16.1, 17.7, 10.3, 3.1, 18.1.
- 2.3.0
  - Bug Fixes
    - Removed DMA_HandleIRQ prototype definition from header file.
    - Added DMA_IRQHandle prototype definition in header file.
- 2.2.5
  - Improvements
    - Added new API DMA_SetupChannelDescriptor to support configuring wrap descriptor.
    - Added wrap support in function DMA_SubmitChannelTransfer.
- 2.2.4
  - Bug Fixes
    - Fixed the issue that macro DMA_CHANNEL_CFER used wrong parameter to calculate DSTINC.

- 2.2.3
  - Bug Fixes
    - Improved DMA driver Deinit function for correct logic order.
  - Improvements
    - Added API DMA_SubmitChannelTransferParameter to support creating head descriptor directly.
    - Added API DMA_SubmitChannelDescriptor to support ping pong transfer.
    - Added macro DMA_ALLOCATE_HEAD_DESCRIPTOR/DMA_ALLOCATE_LINK_DESCRIPTOR to simplify DMA descriptor allocation.
- 2.2.2
  - Bug Fixes
    - Do not use software trigger when hardware trigger is enabled.
- 2.2.1
  - Bug Fixes
    - Fixed Coverity issue.
- 2.2.0
  - Improvements
    - Changed API DMA_SetupDMADescriptor to non-static.
    - Marked APIs below as deprecated.
      - DMA_PrepareTransfer.
      - DMA_Submit transfer.
    - Added new APIs as below:
      - DMA_SetChannelConfig.
      - DMA_PrepareChannelTransfer.
      - DMA_InstallDescriptorMemory.
      - DMA_SubmitChannelTransfer.
      - DMA_SetChannelConfigValid.
      - DMA_DoChannelSoftwareTrigger.
      - DMA_LoadChannelTransferConfig.
- 2.0.1
  - Improvements
    - Added volatile for DMA descriptor member xfercfg to avoid optimization.
- 2.0.0
  - Initial version.

DMIC

The current DMIC driver version is 2.3.2.

- 2.3.2
  - New Features
    - Supported 4 channels in driver.
- 2.3.1
  - Bug Fixes

    - Fixed the issue that DMIC_EnableChannelDma and DMIC_EnableChannelFifo did not clean relevant bits.
- 2.3.0
  - Improvements
    - Added new apis DMIC_ResetChannelDecimator/DMIC_EnableChannelGlobalSync/DMIC_DisableChannelGlobalSync.
- 2.2.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 14.4, 17.7, 10.4, 10.3, 10.8, 14.3.
- 2.2.0
  - Bug Fixes
    - Corrected the usage of feature FSL_FEATURE_DMIC_IO_HAS_NO_BYPASS.
- 2.1.1
  - Improvements
    - Added feature FSL_FEATURE_DMIC_HAS_NO_IOCFG for IOCFG register.
- 2.1.0
  - New Features
    - Added API DMIC_EnbleChannelInterrupt/DMIC_EnbleChannelDma to replace API DMIC_SetOperationMode.
    - Added API DMIC_SetIOCFG and marked DMIC_ConfigIO as deprecated.
    - Added API DMIC_EnableChannelSignExtend to support sign extend feature.
- 2.0.5
  - Improvements
    - Changed some parameters' value of DMIC_FifoChannel API, such as enable, resetn, and trig_level. This is not possible for the current code logic, so it improves the DMIC_FifoChannel logic and fixes incorrect math logic.
- 2.0.4
  - Bug Fixes
    - Fixed the issue that DMIC DMA driver(ver2.0.3) did not support calling DMIC_TransferReceiveDMA in DMA callback as it did before version 2.0.3. But calling DMIC_TransferReceiveDMA in callback is not recommended.
- 2.0.3
  - New Features
  - Supported linked transfer in DMIC DMA driver.
  - Added new API DMIC_EnableChannelFifo/DMIC_DoFifoReset/DMIC_InstallDMADescriptor.
- 2.0.2
  - New Features
    - Supported more channels in driver.
- 2.0.1
  - New Features
    - Added a control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

DMIC_DMA

The current DMIC_DMA driver version is 2.4.0.

- 2.4.0
  - Bug Fixes
    - Fixed the issue that DMIC_TransferAbortReceiveDMA can not disable dmic and dma request issue.
- 2.3.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3.
- 2.3.0
  - Refer DMIC driver change log 2.0.1 to 2.3.0

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.

FLEXCOMM

The current FLEXCOMM driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed typos in FLEXCOMM15_DriverIRQHandler().
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 10.7, 10.8, 11.3, 11.6, 11.8, 11.9, 13.5.
  - Improvements
    - Added instance calculation in FLEXCOMM16_DriverIRQHandler() to align with Flexcomm 14 and 15.
- 2.0.1
  - Improvements
    - Added more IRQHandler code in drivers to adapt new devices.
- 2.0.0
  - Initial version.

FLEXSPI

The current FLEXSPI driver version is 2.6.0.

- 2.6.0
  - New Features
    - Added new API to set AHB memory-mapped flash base address.
    - Added support of DLLxCR[REFPHASEGAP] bit field, it is recommended to set it as 0x2 if DLL calibration is enabled.
- 2.5.1

  - Bugfixes
    - Fixed handling of W1C bits in the INTR register
    - Removed FIFO resets from FLEXSPI_CheckAndClearError
    - FLEXSPI_TransferBlocking is observing IPCMDDONE and then fetches the final status of the transfer
    - Fixed issue that FLEXSPI2_DriverIRQHandler not defined.
- 2.5.0
  - Improvements
    - Supported word un-aligned access for write/read blocking/non-blocking API functions.
    - Fixed dead loop issue in DLL update function when using FRO clock source.
    - Fixed violations of the MISRA C-2012 Rule 10.3.
- 2.4.0
  - Improvements
    - Isolated IP command parallel mode and AHB command parallel mode using feature MACRO.
    - Supported new column address shift feature for external memory.
- 2.3.5
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 14.2.
- 2.3.4
  - Bug Fixes
    - Updated flexspi_config_t structure and FlexSPI_Init to support new feature FSL_FEATURE_FLEXSPI_HAS_NO_MCR0_CONBINATION.
- 2.3.3
  - Bug Fixes
    - Removed feature FSL_FEATURE_FLEXSPI_DQS_DELAY_PS for DLL delay setting. Changed to use feature FSL_FEATURE_FLEXSPI_DQS_DELAY_MIN to set slave delay target as 0 for DLL enable and clock frequency higher than 100MHz.
- 2.3.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 8.4, 8.5, 10.1, 10.3, 10.4, 11.6 and 14.4.
- 2.3.1
  - Bug Fixes
    - Wait for bus to be idle before using it as access to external flash with new setting in FLEXSPI_SetFlashConfig() API.
    - Fixed the potential buffer overread and Tx FIFO overwrite issue in FLEXSPI_WriteBlocking.
- 2.3.0
  - New Features
    - Added new API FLEXSPI_UpdateDllValue for users to update DLL value after updating flexspi root clock.
    - Corrected grammatical issues for comments.
    - Added support for new feature FSL_FEATURE_FLEXSPI_DQS_DELAY_PS in DLL configuration.
- 2.2.2

- Bug Fixes
  - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3 and 10.4.
  - Updated _flexspi_command from named enumerator into anonymous enumerator.
- 2.2.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.8, 11.9, 14.4, 15.7, 16.4, 17.7, 7.3.
    - Fixed IAR build warning Pe167.
    - Fixed the potential buffer overwrite and Rx FIFO overread issue in FLEXSPI_ReadBlocking.
- 2.2.0
  - Bug Fixes
    - Fixed flag name typos: kFLEXSPI_IpTxFifoWatermarkEmpltyFlag to kFLEXSPI_IpTxFifoWatermarkEmptyFlag; kFLEXSPI_IpCommandExcutionDoneFlag to kFLEXSPI_IpCommandExecutionDoneFlag.
    - Fixed comments typos such as sequencen->sequence, levle->level.
    - Fixed FLSHCR2[ARDSEQID] field clean issue.
    - Updated flexspi_config_t structure and FlexSPI_Init to support new feature FSL_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.

FLEXSPI DMA Driver

The current FLEXSPI DMA driver version is 2.2.1.

- 2.2.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.8.
- 2.2.0
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 Rule 10.1, 10.3.
  - New Features
    - Updated name of FLEXSPI_TransferGetTransferCountDMA API.
- 2.1.1
  - New Features
    - Updated driver to support feature FSL_FEATURE_FLEXSPI_DMA_MULTIPLE_DES.
- 2.1.0
  - Bug Fixes
    - Updated enumaration flexspi_dma_transfer_nsize_t and remove the unsupported items.
  - New Features
    - Updated driver for deprecating the multiple linked descriptors inside FLEXSPI_TransferDMA, only up to one linked descriptor is needed according to hardware update.

- 2.0.0
  - Initial version.

FMEAS

The current FMEAS driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - MISRA C-2012 issues fixed: rule 10.4, rule 10.8.
- 2.1.0
  - Updated "FMEAS_GetFrequency","FMEAS_StartMeasure","FMEAS_IsMeasureComplete" API and add definition to match ASYNC_SYSCON.
- 2.0.0
  - Initial version ported from LPCOpen.

GDMA

The current GDMA driver version is 2.0.3.

- 2.0.3
  - Bug Fixes
    - Fixed MISRA C-2012 violation.
- 2.0.2
  - Improvements
    - Changed to use FSL_FEATURE_GDMA_CHANNEL_NUM defined by feature header.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA C-2012 violation.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.1.7.

- 2.1.7
  - Improvements
    - Enhanced GPIO_PinInit to enable clock internally.
- 2.1.6
  - Bug Fixes
    - Clear bit before set it within GPIO_SetPinInterruptConfig() API.
- 2.1.5
  - Bug Fixes

    - Fixed violations of the MISRA C-2012 rules 3.1, 10.6, 10.7, 17.7.
- 2.1.4
  - Improvements
    - Added API GPIO_PortGetInterruptStatus to retrieve interrupt status for whole port.
    - Corrected typos in header file.
- 2.1.3
  - Improvements
    - Updated "GPIO_PinInit" API. If it has DIRCLR and DIRSET registers, use them at set 1 or clean 0.
- 2.1.2
  - Improvements
    - Removed deprecated APIs.
- 2.1.1
  - Improvements
    - API interface changes:
      - Refined naming of APIs while keeping all original APIs, marking them as deprecated. Original APIs will be removed in next release. The mainin change is updating APIs with prefix of _PinXXX() and _PorortXXX
- 2.1.0
  - New Features
    - Added GPIO initialize API.
- 2.0.0
  - Initial version.

IMU

The Current IMU driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - Fix MISRA C-2012 violations.
    - Fixed IMU_GetStatusFlags bug that returns wrong RX FIFO status.
- 2.1.0
  - Improvements:
    - Updated API prototype, remove CIU2_Type from parameters.
- 2.0.0
  - Initial version.

INPUTMUX

The current INPUTMUX driver version is 2.0.7.

- 2.0.7
  - Improvements

    - Release peripheral from reset if necessary in init function.
- 2.0.6
  - Bug Fixes
    - Fixed the documentation wrong in API INPUTMUX_AttachSignal.
- 2.0.5
  - Bug Fixes
    - Fixed build error because some devices has no sct.
- 2.0.4
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rule 10.4, 12.2 in INPUTMUX_EnableSignal() function.
- 2.0.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.4, 10.7, 12.2.
- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.4, 12.2.
- 2.0.1
  - Support channel mux setting in INPUTMUX_EnableSignal().
- 2.0.0
  - Initial version.

LCDIC

The current LCDIC driver version is 2.1.0.

- 2.1.0
  - New Features
    - Add seperate APIs for send and receive data in non-blocking way.
  - Others
    - Return error status when sending or receiving data larger than 0x40000, current driver doesn't support this.
- 2.0.3
  - Bug Fixes
    - Fixed potential issue that clock may not be send out when sending data array.
- 2.0.2
  - Bug Fixes
    - Fixed build error with MDK 5.37.
- 2.0.1
  - Bug Fixes
    - Added delay after setting LCDIC_EN to make sure LCDIC is out of reset.
- 2.0.0
  - Initial version.

LCDIC_DMA

The current LCDIC DMA driver version is 2.1.0.

- 2.1.0
  - New Features
    - Add seperate APIs for send and receive data.
  - Others
    - Return error status when sending or receiving data larger than 0x40000, current driver doesn't support this.
- 2.0.0
  - Initial version.

MRT

The current MRT driver version is 2.0.4.

- 2.0.4
  - Improvements
    - Don't reset MRT when there is not system level MRT reset functions.
- 2.0.3
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.1 and 10.4.
    - Fixed the wrong count value assertion in MRT_StartTimer API.
- 2.0.2
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rule 10.4.
- 2.0.1
  - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

OSTIMER

The current OSTIMER driver version is 2.2.2.

- 2.2.2
  - Improvements
    - Support devices with different OSTIMER instance name.
- 2.2.1
  - Improvements
    - Release peripheral from reset if necessary in init function.
- 2.2.0
  - Improvements
    - Move the PMC operation out of the OSTIMER driver to board specific files.

        - Added low level APIs to control OSTIMER MATCH and interrupt.
- 2.1.2
    - Bug Fixes
        - Fixed MISRA-2012 rule 10.8.
- 2.1.1
    - Bug Fixes
        - removes the suffix 'n' for some register names and bit fields' names
    - Improvements
        - Added HW CODE GRAY feature supported by CODE GRAY in SYSCTRL register group.
- 2.1.0
    - Bug Fixes
        - Added a workaround to fix the issue that no interrupt was reported when user set smaller period.
        - Fixed violation of MISRA C-2012 rule 10.3 and 11.9.
    - Improvements
        - Added return value for the two APIs to set match value.
            - OSTIMER_SetMatchRawValue
            - OSTIMER_SetMatchValue
- 2.0.3
    - Bug Fixes
        - Fixed violation of MISRA C-2012 rule 10.3, 14.4, 17.7.
- 2.0.2
    - Improvements
        - Added support for OSTIMER0
- 2.0.1
    - Improvements
        - Removed the software reset function out of the initialization API.
        - Enabled interrupt directly instead of enabling deep sleep interrupt. Users need to enable the deep sleep interrupt in application code if needed.
- 2.0.0
    - Initial version.

PINT

The current PINT driver version is 2.1.13.

- 2.1.13
    - Improvements
        - Added instance array for PINT to adapt more devices.
        - Used release reset instead of reset PINT which may clear other related registers out of PINT.
- 2.1.12
    - Bug Fixes

    - Fixed coverity issue.
- 2.1.11
  - Bug Fixes
    - Fixed MISRA C-2012 rule 10.7 violation.
- 2.1.10
  - New Features
    - Added the driver support for MCXN10 platform with combined interrupt handler.
- 2.1.9
  - Bug Fixes
    - Fixed MISRA-2012 rule 8.4.
- 2.1.8
  - Bug Fixes
    - Fixed MISRA-2012 rule 10.1 rule 10.4 rule 10.8 rule 18.1 rule 20.9.
- 2.1.7
  - Improvements
    - Added fully support for the SECPINT, making it can be used just like PINT.
- 2.1.6
  - Bug Fixes
    - Fixed the bug of not enabling common pint clock when enabling security pint clock.
- 2.1.5
  - Bug Fixes
    - Fixed issue for MISRA-2012 check.
      - Fixed rule 10.1 rule 10.3 rule 10.4 rule 10.8 rule 14.4.
    - Changed interrupt init order to make pin interrupt configuration more reasonable.
- 2.1.4
  - Improvements
    - Added feature to control distinguish PINT/SECPINT relevant interrupt/clock configurations for PINT_Init and PINT_Deinit API.
    - Swapped the order of clearing PIN interrupt status flag and clearing pending NVIC interrupt in PINT_EnableCallback and PINT_EnableCallbackByIndex function.
    - Bug Fixes
      - Fixed build issue caused by incorrect macro definitions.
- 2.1.3
  - Bug fix:
    - Updated PINT_PinInterruptClrStatus to clear PINT interrupt status when the bit is asserted and check whether was triggered by edge-sensitive mode.
    - Write 1 to IST corresponding bit will clear interrupt status only in edge-sensitive mode and will switch the active level for this pin in level-sensitive mode.
    - Fixed MISRA c-2012 rule 10.1, rule 10.6, rule 10.7.
    - Added FSL_FEATURE_SECPINT_NUMBER_OF_CONNECTED_OUTPUTS to distinguish IRQ relevant array definitions for SECPINT/PINT on lpc55s69 board.
    - Fixed PINT driver c++ build error and remove index offset operation.
- 2.1.2
  - Improvement:
    - Improved way of initialization for SECPINT/PINT in PINT_Init API.

- 2.1.1
  - Improvement:
    - Enabled secure pint interrupt and add secure interrupt handle.
- 2.1.0
  - Added PINT_EnableCallbackByIndex/PINT_DisableCallbackByIndex APIs to enable/disable callback by index.
- 2.0.2
  - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
  - Bug fix:
    - Updated PINT driver to clear interrupt only in Edge sensitive.
- 2.0.0
  - Initial version.

POWERQUAD

The current POWERQUAD driver version is 2.2.0.

- 2.2.0
  - New Features
    - Added new API PQ_Arctan2Fixed.
- 2.1.1
  - Bug Fixes
    - Remove PQ_WaitDone from PQ_ArctanFixed and PQ_ArctanhFixed because it is unnecessary.
- 2.1.0
  - Improvements
    - Fixed typo issue for biquad related function name.
    - Changed operator from "%" into "&" to reduce heavy cycle for biquad functions.
- 2.0.5
  - Improvements
    - Added a note in driver for FIR that powerquad has a hardware limitation, when using it for FIR increment calculation, the address of pSrc needs to be a continuous address.
- 2.0.4
  - Improvements
    - Supported the platforms which don’t have PowerQuad clock and reset control.
- 2.0.3
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 8.4, 10.1, 10.3, 10.4, 10.6, and so on.
- 2.0.2
  - Bug Fixes
    - Fixed array size issue in fsl_powerquad_data.h file.
    - Fixed vector function pipeline issue.
- 2.0.1

- Bug Fixes
  - Fixed build error in C++ mode.
- 2.0.0
  - Initial version.

RTC

The current RTC driver version is 2.2.0.

- 2.2.0
  - New Features
    - Created new APIs for the RTC driver.
      - RTC_EnableSubsecCounter
      - RTC_GetSubsecValue
- 2.1.3
  - Bug Fixes
    - Fixed issue that RTC_GetWakeupCount may return wrong value.
- 2.1.2
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.1, 10.4 and 10.7.
- 2.1.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3 and 11.9.
- 2.1.0
  - Bug Fixes
    - Created new APIs for the RTC driver.
      - RTC_EnableTimer
      - RTC_EnableWakeUpTimerInterruptFromDPD
      - RTC_EnableAlarmTimerInterruptFromDPD
      - RTC_EnableWakeupTimer
      - RTC_GetEnabledWakeupTimer
      - RTC_SetSecondsTimerMatch
      - RTC_GetSecondsTimerMatch
      - RTC_SetSecondsTimerCount
      - RTC_GetSecondsTimerCount
    - deprecated legacy APIs for the RTC driver.
      - RTC_StartTimer
      - RTC_StopTimer
      - RTC_EnableInterrupts
      - RTC_DisableInterrupts
      - RTC_GetEnabledInterrupts
- 2.0.0
  - Initial version.

SCTIMER

The current SCTimer driver version is 2.5.1.

- 2.5.1
  - Bug Fixes
    - Fixed bug in SCTIMER_SetupCaptureAction: When kSCTIMER_Counter_H is selected, events 12-15 and capture registers 12-15 CAPn_H field can't be used.
- 2.5.0
  - Improvements
    - Add SCTIMER_GetCaptureValue API to get capture value in capture registers.
- 2.4.9
  - Improvements
    - Supported platforms which don't have system level SCTIMER reset.
- 2.4.8
  - Bug Fixes
    - Fixed the issue that the SCTIMER_UpdatePwmDutycycle() can't writes MATCH_H bit and RELOADn_H.
- 2.4.7
  - Bug Fixes
    - Fixed the issue that the SCTIMER_UpdatePwmDutycycle() can't configure 100% duty cycle PWM.
- 2.4.6
  - Bug Fixes
    - Fixed the issue where the H register was not written as a word along with the L register.
    - Fixed the issue that the SCTIMER_SetCOUNTValue() is not configured with high 16 bits in unify mode.
- 2.4.5
  - Bug Fixes
    - Fix SCT_EV_STATE_STATEMSKn macro build error.
- 2.4.4
  - Bug Fixes
    - Fix MISRA C-2012 issue 10.8.
- 2.4.3
  - Bug Fixes
    - Fixed the wrong way of writing CAPCTRL and REGMODE registers in SCTIMER_SetupCaptureAction.
- 2.4.2
  - Bug Fixes
    - Fixed SCTIMER_SetupPwm 100% duty cycle issue.
- 2.4.1
  - Bug Fixes
    - Fixed the issue that MATCHn_H bit and RELOADn_H bit could not be written.
- 2.4.0
- 2.3.0
  - Bug Fixes

        - Fixed the potential overflow issue of pulseperiod variable in SCTIMER_SetupPwm/SCTIMER_UpdatePwmDutycycle API.
        - Fixed the issue of SCTIMER_CreateAndScheduleEvent API does not correctly work with 32 bit unified counter.
        - Fixed the issue of position of clear counter operation in SCTIMER_Init API.
    - Improvements
        - Update SCTIMER_SetupPwm/SCTIMER_UpdatePwmDutycycle to support generate 0% and 100% PWM signal.
        - Add SCTIMER_SetupEventActiveDirection API to configure event activity direction.
        - Update SCTIMER_StartTimer/SCTIMER_StopTimer API to support start/stop low counter and high counter at the same time.
        - Add SCTIMER_SetCounterState/SCTIMER_GetCounterState API to write/read counter current state value.
        - Update APIs to make it meaningful.
            - SCTIMER_SetEventInState
            - SCTIMER_ClearEventInState
            - SCTIMER_GetEventInState
- 2.2.0
    - Improvements
        - Updated for 16-bit register access.
- 2.1.3
    - Bug Fixes
        - Fixed the issue of uninitialized variables in SCTIMER_SetupPwm.
        - Fixed the issue that the Low 16-bit and high 16-bit work independently in SCTIMER driver.
    - Improvements
        - Added an enumerable macro of unify counter for user.
            - kSCTIMER_Counter_U
        - Created new APIs for the RTC driver.
            - SCTIMER_SetupStateLdMethodAction
            - SCTIMER_SetupNextStateActionwithLdMethod
            - SCTIMER_SetCOUNTValue
            - SCTIMER_GetCOUNTValue
            - SCTIMER_SetEventInState
            - SCTIMER_ClearEventInState
            - SCTIMER_GetEventInState
        - Deprecated legacy APIs for the RTC driver.
            - SCTIMER_SetupNextStateAction
- 2.1.2
    - Bug Fixes
        - MISRA C-2012 issue fixed: rule 10.3, 10.4, 10.6, 10.7, 11.9, 14.2 and 15.5.
- 2.1.1
    - Improvements
        - Updated the register and macro names to align with the header of devices.
- 2.1.0

  - Bug Fixes
    - Fixed issue where SCT application level Interrupt handler function is occupied by SCT driver.
    - Fixed issue where wrong value for INSYNC field inside SCTIMER_Init function.
    - Fixed issue to change Default value for INSYNC field inside SCTIMER_GetDefaultConfig.
- 2.0.1
  - New Features
    - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

UTICK

The current UTICK driver version is 2.0.5.

- 2.0.5
  - Improvements
    - Improved for SOC RW610.
- 2.0.4
  - Bug Fixes
    - Fixed compile fail issue of no-supporting PD configuration in utick driver.
- 2.0.3
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 8.4, 14.4, 17.7
- 2.0.2
  - Added new feature definition macro to enable/disable power control in drivers for some devices have no power control function.
- 2.0.1
  - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

WWDT

The current WWDT driver version is 2.1.9.

- 2.1.9
  - Bug Fixes
    - Fixed violation of the MISRA C-2012 rule 10.4.
- 2.1.8
  - Improvements
    - Updated the "WWDT_Init" API to add wait operation. Which can avoid the TV value read by CPU still be 0xFF (reset value) after WWDT_Init function returns.

- 2.1.7
  - Bug Fixes
    - * Fixed the issue that the watchdog reset event affected the system from PMC.
    - * Fixed the issue of setting watchdog WDPROTECT field without considering the backwards compatibility.
    - * Fixed the issue of clearing bit fields by mistake in the function of WWDT_ClearStatusFlags.
- 2.1.5
  - Bug Fixes
    - * deprecated a unusable API in WWWDT driver.
      - · WWDT_Disable
- 2.1.4
  - Bug Fixes
    - * Fixed violation of the MISRA C-2012 rules Rule 10.1, 10.3, 10.4 and 11.9.
    - * Fixed the issue of the inseparable process interrupted by other interrupt source.
      - · WWDT_Init
- 2.1.3
  - Bug Fixes
    - * Fixed legacy issue when initializing the MOD register.
- 2.1.2
  - Improvements
    - * Updated the "WWDT_ClearStatusFlags" API and "WWDT_GetStatusFlags" API to match QN9090. WDTOF is not set in case of WD reset. Get info from PMC instead.
- 2.1.1
  - New Features
    - * Added new feature definition macro for devices which have no LCOK control bit in MOD register.
    - * Implemented delay/retry in WWDT driver.
- 2.1.0
  - Improvements
    - * Added new parameter in configuration when initializing WWDT module. This parameter, which must be set, allows the user to deliver the WWDT clock frequency.
- 2.0.0
  - Initial version.

2 Middleware Change Log

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

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: d0efd9ef7ba08e54b46b1060e2b4629a49 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: 0a0452b2c39bdd91e252aef045c115f88f6ca7 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: 3fe8d2fc43a9b69f7ed28c63d44a7744f9c 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: 239918ccc173cb2c2a62f41a40fd893f57f 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 ENET_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: 124dc0a64ef5d7c14a27e3115e5888df65 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: 7ec4e9be304e7f8953740f10b2c810a292 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_NUM 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: 0056522cc974d2be2005c324f37187b5b3 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: c385f31076b27efb8ee37f00cb5568783a 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: 555812dcec38c9a2ef1ef9b31816291549 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 N-`

        `O_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: 1bb6e7f52de1cd86be0eed31e3 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.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.10.1.

- 2.10.1
  - Improvement:
    - Host cdc acm
      - Add link state detection in DHCP and DNS processing.
      - Unify and simplify application processing log.
      - Add static IP supporting.
    - Update USB device controller drivers to support Zephyr.
    - Upadte default USB OR register settings to pass USB certification/compliance.

  - Bug fixes:
    - update device echi driver to fix the issue that EPCOMPLERE bit is not cleared after transfer is canceled.
    - update usb host hub class to fix the wrong hub descriptor class code.
- 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_DEVIC-E_CONFIG_RETURN_VALUE_CHECK.
    - Change the header file including order for usb.h header.

    - Update the USB host audio class driver to fix the wrong output log.
    - Add the workaround on dev_hid_mouse_bm case for the errata TN00071.
    - Enable ROOT2 macro in USB device stack.
    - Use an unified definiton for the base address of RTxxxx platforms.
- 2.8.3
  - Improvement:
    - Update the EHCI controller driver to support the address convert for TCM.
    - Update the USB host EHCI controller driver to make sure the mutual exclusion access under multiple tasks' environment.
- 2.8.2
  - Improvement:
    - Fix noise issue of UAC 3.1, UAC 5.1, UAC 7.1 on usb audio speaker demo.
    - Fix the issue that incorrect PC behavior when ejecting USB MSC devices.
    - Update the EHCI controller driver to support RW610 that does not reply on PHY driver, especially for low power feature.
    - Update the USB_HostHelperParseAlternateSetting to fix the wrong interface parse.
    - Update dev_composite_hid_audio_unified_bm demo to suppport independent mute/unmute and volume control.
- 2.8.1
  - Improvement:
    - update USB audio demos to use audio component (components/audio).
    - Add the checking of function call return value.
    - Add audio multiple channels demo (usb_device_composite_audio_multi_ch_unified) on RT600 audio board.
    - Fix audio noise on sync mode and improve overflow/underflow checking method.
    - Support UAC 3.1, 5.1 and 7.1 on audio speaker demo.
    - Set USB device CDC demo not to depend on DTR setting from host.
    - Support MCUX toolchain on some RTxxxx platforms.
- 2.8.0
  - Improvement:
    - Fix the USB device stack vulnerability issues.
    - Update the audio PLL and FRO adjustment codes for audio examples in RTxxx, LPC54xxx and LPC55xxx.
    - Improve the USB PD AMS collision avoidance.
    - Improve IP3511 controller driver's dedicated ram allocation.
    - Change the USB_DATA_ALIGN_SIZE to 4 because the controller driver uses the dedicated RAM to do memcpy.
  - New features:
    - Enable USB host audio recorder demo for mutilple boards.
- 2.7.0
  - Improvement:
    - Use new feeback solution and low latency playback for usb device speaker demo and unified demos. Add underflow and overflow protection.
    - Optimize hard code for usb audio demos.
    - Update Unconstrained Power field in the Sink Capabilities Message according to the

    external power state.
    - Fix CVE-2021-38258 and CVE-2021-38260
  - New features:
    - Enable USB host video demo for mutilple boards.
    - Enable USB device MTP demo for mutilple boards.
    - Add PPS message to usb pd stack.
- 2.6.1
  - Improvement:
    - rename sdcard as disk for all of sdcard demos. For ramdisk demos, they are not changed.
    - add wrapper for all of disk demos to support emmc.
- 2.6.0
  - Improvement:
    - Added more ufi event to support dynamic sdcard capacity.
    - Passed MISRA-2012 mandatory and required rules.
      - Except rule 17.2 in host hub and otg stack.
      - Except rule 5.1, rule 5.4, rule 21.1 and rule 21.2.
    - Re-implemented USB components and supported NPW.
    - Improved IP3511 controller driver's cancelling transfer function.
    - Enabled the audio2.0 defaultly for device audio demos.
    - Enabled the host audio2.0 function in host audio class driver and host audio speaker demo.
  - New features:
    - enable two USB controllers in one USB host mouse demo which named as host_hid_mouse_dual.
    - enable UAC 5.1 for usb device audio speaker demo.
- 2.5.0
  - Improvement:
    - Integrated sdk components (OSA, Timer, GPIO and serial_manager) to USB stack and demos.
    - Improved the ip3511 driver throughput.
    - Improved audio initialization codes after SDK audio drivers update.
    - Improved auido to support the audio2.0 in win10.
    - Add one "enumeration fail" callback event to host stack.
- 2.4.2
  - Improvement:
    - Put the USB controller data and transfer buffer to noncache section, removed the setting that sets the whole ocram and sdram as noncached.
    - Separated composite audio examples' channel,sample rate,format parameters from commom macro to in dedicated macro and out dedicated macro.
    - replaced USB_PrepareData with USB_AudioRecorderGetBuffer.
- 2.4.1
  - New features:
    - Added enumeration fail callback to host stack when the attached device's enumeration failed.
- 2.4.0
  - Improvement:

    - Device Charger Detection (DCD) software architecture was refactored.
  - New features:
    - Enabled Device Charger Detection (DCD) on RT1060.
    - Enabled Device Charger Detection on RT600.
    - Enabled host battery charger function on RT600.
- 2.3.0
  - New features:
    - Added host video camera support. example: usb_host_video_camera
    - Added a new device example. example: usb_device_composite_cdc_hid_audio_unified
- 2.2.0
  - New features:
    - Added device DFU support.
    - Supported OM13790DOCK on LPCXpresso54018.
    - Added multiple logical unit support in msc class driver, updated usb_device_lba_information_struct_t to support this.
    - Supported multiple transfers for host ISO on IP3516HS.
  - Bug fixes:
    - Fixed device ip3511 prime data length than maxpacket size issue.
    - Initialized interval attribute in usb_device_endpoint_struct_t/usb_device_endpoint_init_struct_t.
    - Removed unnecessary header file in device CDC class driver, removed unnecessary usb_echo, and added DEBUG macro for necessary usb_echo in device CDC class driver.
    - Fixed device IP3511HS unfinished interrupt transfer missing issue.
- 2.1.0
  - New features:
    - Added host RNDIS support. example: lwip_dhcp_usb
    - Enabled USB 3.0 support on device stack.
    - Power Delivery feature: Added OM13790HOST support; Added auto policy feature; Printed e-marked cable information;
- 2.0.1
  - Bug fixes:
    - Fixed some USB issues: Fixed MSC CV test failed in MSC examples.
    - Changed audio codec interfaces.
- 2.0.0
  - New features:
    - PTN5110N support.
  - Bug fix:
    - Added some comments, fixed some minor USB issues.
- 1.9.0
  - New features:
    - Examples:
      - usb_pd_alt_mode_dp_host
- 1.8.2
  - Updated license.
- 1.8.1

  - Bug fix:
    - Verified some hardware issues, support aruba_flashless.
- 1.8.0
  - New features:
    - Examples:
      - usb_device_composite_cdc_vcom_cdc_vcom
      - usb_device_composite_hid_audio_unified
      - usb_pd_sink_battery
      - Changed usb_pd_battery to usb_pd_charger_battery.
  - Bug fix:
    - Code clean up, removed some irrelevant code.
- 1.7.0
  - New features:
    - USB PD stack support.
  - Examples:
    - usb_pd
    - usb_pd_battery
    - usb_pd_source_charger
- 1.6.3
  - Bug fix: -IP3511_HS driver control transfer sequence issue, enabled 3511 ip cv test.
- 1.6.2
  - New features:
    - Multi instance support.
- 1.6.1
  - New features:
  - Changed the struct variable address method for device_video_virtual_camera and host_phdc-_manager.
- 1.6.0
  - New features:
    - Supported Device Charger Detect feature on usb_device_hid_mouse.
- 1.5.0
  - New features:
    - Supported controllers
      - OHCI (Full Speed, Host mode)
      - IP3516 (High Speed, Host mode)
      - IP3511 (High Speed, Device mode)
    - Examples:
      - usb_lpm_device_hid_mouse
      - usb_lpm_device_hid_mouse_lite
      - usb_lpm_host_hid_mouse
- 1.4.0
  - New features:
    - Examples:
      - usb_device_hid_mouse/freertos_static
      - usb_suspend_resume_device_hid_mouse_lite

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

- · VIDEO

∗ Examples:

- · usb_device_audio_generator
- · usb_device_audio_speaker
- · usb_device_ccid_smart_card
- · usb_device_cdc_vcom
- · usb_device_cdc_vnic
- · usb_device_composite_cdc_msc
- · usb_device_composite_hid_audio
- · usb_device_composite_hid_mouse_hid_keyboard
- · usb_device_hid_generic
- · usb_device_hid_mouse
- · usb_device_msc_ramdisk
- · usb_device_msc_sdcard
- · usb_device_phdc_weighscale
- · usb_device_video_flexio_ov7670
- · usb_device_video_virtual_camera
- · usb_host_audio_speaker
- · usb_host_cdc
- · usb_host_hid_generic
- · usb_host_hid_mouse
- · usb_host_hid_mouse_keyboard
- · usb_host_msd_command
- · usb_host_msd_fatfs
- · usb_host_phdc_manager
- · usb_keyboard2mouse
- · usb_pin_detect_hid_mouse

3 Component Change Log

CODEC

The current codec common driver version is 2.3.1.

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

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

WM8904

The current wm8904 driver version is 2.5.1.

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

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

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