MCUXpresso SDK Release Notes Supporting mekmimx8qx

Change Logs

Contents

Contents

Driver Change Log

CLOCK

The current CLOCK driver version is 2.3.1

- 2.3.1
  - Bug Fixes
    - Fixed misra rule 20.7, rule 14.4, rule 11.9.
- 2.3.0
  - New feature:
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.2.0
  - New feature
    - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.1.0
  - New Features:
    - Update clock_ip_name_t, adding LPCG base information.
    - Add LPCG configure API CLOCK_ConfigLPCG/CLOCK_SetLpcgGate.
    - Update CLOCK_EnableClockExt/CLOCK_DisableClock API, adding LPCG gate control.
- 2.0.1
  - some minor fixes.
- 2.0.0
  - initial version.

LPI2C_CMSIS

Current LPI2C_CMSIS driver version is 2.0

- 2.0
  - Initial version.

LPSPI_CMSIS

Current LPSPI_CMSIS driver version is 2.2

- 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.

ASMC

The current ASMC driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed issue for MISRA-2012 rule 10.1, rule 10.4.
- 2.0.0
  - Initial version.

CACHE

The current CACHE driver version is 2.0.4.

- 2.0.4
  - Bug Fixes
    - Updated full name for lmem driver.
    - Fixed doxygen issue.
- 2.0.3
  - Bug Fixes
    - Fixed violation of MISRA C-2012 Rule 10.4 and 14.4.
- 2.0.2
  - Improvements
    - Moved CLCR register configuration out of the while loop, it's unnecessary to repeat this operation.
- 2.0.1

  - Bug Fixes
    - Fixed the over-4KB-size maintenance issue in invalidate/clean/clean&invalidate by range APIs.
- 2.0.0
  - Initial version.

CI_PI

The current CI_PI driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules: 10.1, 10.4, 10.8, 14.4, 17.7.
- 2.0.1
  - Other Changes
    - Update to use new C0 version header files.
- 2.0.0
  - Initial version.

COMMON

The current COMMON driver version is 2.2.4.

- 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 let result be more close 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.

EDMA

The current eDMA driver version is 2.2.2.

- 2.2.2
  - Bug Fixes
    - Fixed the issue of EDMA_SubmitTransfer return busy when calling EDMA_EnableChannelInterrupts before submit transfer.
    - Fixed violations of MISRA C-2012 rule 10.4, 10.1, 9.2, 10.4, 10.6, 14.4, 10.7, 14.3, 11.6.
- 2.2.1
  - Improvements
    - Removed channel MUX reset from EDMA_ResetChannel, since channel mux should be constant while channel is alive.
- 2.2.0
  - Improvements

        * Added new API EDMA_SetChannelMux to support channel mux feature.
        * Added new API EDMA_PrepareTransferConfig to expose paramters source offset and destination offset.
        * Exposed EDMA_InstallTCD function to application.
        * Added source/destination address alignment check.
- 2.1.1
  - Improvements
    * Added 8bytes transfer width feature support in driver.
- 2.1.0
  - Bug Fixes
    * Added const type for parameter configuration in EDMA_SubmitTransfer and EDMA_-HandleTransferConfig API.
    * Added configurations for srcAddr and destAddr in EDMA_PrepareTransfer API.
- 2.0.2
  - Improvements
    * Updated eDMA driver to support MP_CR bit GMRC.
    * Updated eDMA instance name for i.MX 8QM.
    * Used instance number as factor to calculate channel number for different instance instead of hard code.
- 2.0.1
  - New Features
    * Added control macro to enable/disable the CLOCK code in current driver.
    * Added s_EDMAEnabledChannel to record enabled channel to merge all the channel IRQ handler into driver IRQ handler.
    * Added feature macro for bits EMI and EBW in MP_CSR.
  - Improvements
    * Removed all the separated channel IRQ handler in DMA driver.
- 2.0.0
  - Initial version.

PRG

The current PRG driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    * Fixed the violations of MISRA 2012 rules: 10.6.
- 2.0.0
  - Initial version.

DPU

The current DPU driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - Fixed MISRA-2012 issues: 3.1, 8.4, 8.6, 10.1, 10.3, 10.4, 10.8, 12.2, 14.4, 15.7, 17.7, 20.7.
- 2.1.0
  - Improvements
    - Made the fetch unit burst length configurable.
- 2.0.0
  - Initial version.

DPU IRQSTEER

The current DPU IRQSTEER driver version is 2.0.0.

- 2.0.0
  - Initial version.

ENET

The current ENET driver version is 2.3.0.

- 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_RxBuffer-Interrupt, from kENET_TxByteInterrupt to kENET_TxBufferInterrupt.
    * Deleted unnecessary parameters in ENET handler.
- 2.0.0
  - Initial version.

ESAI

The current ESAI driver version is 2.1.0.

- 2.1.0
  - Bug Fixes

        * Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.7, 11.6, 11.9, 14.4, 16.4, 17.7, 8.4, 8.5.
- 2.0.1
  - New Features
    * Added a control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

FLEXCAN

The current FLEXCAN driver version is 2.5.2.

- 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.

FLEXSPI

The current FLEXSPI driver version is 2.2.2.

- 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 FLEX-SPI_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 FL-EXSPI_TransferHandleIRQ.
    - Updated eDMA API FLEXSPI_TransferEDMA.
- 2.0.0
  - Initial version.

FTM

The current FTM driver version is 2.2.3.

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

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

GPT

The current GPT driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.8, 17.7.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.0.3.

- 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.

INTMUX

The current INTMUX driver version is 2.0.3.

- 2.0.3
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 17.7 and 18.1.
- 2.0.2
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.3, 10.4, and 14.4.
- 2.0.1
  - Improvements
    - Added weak function implementations of INTMUX1_x_DriverIRQHandler. x ranges from 0 to 7 to support 8 channels.
- 2.0.0
  - Initial version.

IRQSTEER

The current IRQSTEER driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed the violation of MISRA C-2012 Rules: 10.1, 10.4, 10.7, 10.8, 12.2, 17.7, 20.7.
  - New Features
    - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

ISI

The current ISI driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules: 3.1, 5.6, 5.7, 10.1, 10.3, 10.4, 10.8, 12.2, 13.2, 14.4, 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.

LDB (LVDS Display Bridge)

The current LDB driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 8.4, 10.1, 10.4, 10.7, 12.2, 14.4, 17.7, 20.7.
- 2.0.0
  - Initial version.

LPADC

The current LPADC driver version is 2.2.1.

- 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 2.1.11.

- 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.

LPIT

The current LPIT driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed rules, containing: rule-10.3, rule-14.4, rule-15.5.
- 2.0.0
  - Initial version.

LPSPI

The current LPSPI driver version is 2.0.5.

- 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.

LPUART

The current LPUART driver version is 2.3.0.

- 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.

CSI2RX

The current CSI2RX driver version is 2.0.3.

- 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 2.0.4.

- 2.0.4
  - Bug Fixes
    - Fixed MISRA C-2012 issues: 10.1, 10.3, 10.4, 10.4, 10.6, 10.7, 10.8, 11.3, 11.8, 12.2, 14.4, 16.4, 17.7.
- 2.0.3
  - Improvement
    - Updated for combo phy header file.
- 2.0.2
  - New Features
    - Supported sending separate DSI command from TX data array.
  - Bug Fixes
    - Disabled all interrupts in DSI_Init.
- 2.0.1
  - Improvements
    - Updated to support the DPHY which does not have internal DPHY PLL.
- 2.0.0
  - Initial version.

MU

The Current MU driver version is 2.0.5.

- 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.

PRG

The current PRG driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed the violations of MISRA 2012 rules 10.6.
- 2.0.0
  - Initial version.

RGPIO

The current RGPIO driver version is 2.0.2.

- 2.0.2
  - Bug fix
    - MISRA C-2012 issue fixed.
      - Fix rules, containing: rule-10.3, rule-14.4, rule-15.5.
- 2.0.1
  - API Interface Change:
    - Refined naming of API while keep all original APIs with marking them as deprecated. The original API will be removed in the next release. The main change is to update API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.3.1

- 2.3.1
  - Bug Fixes
    - Corrected the peripheral name in function SAI0_DriverIRQHandler.
    - Fixed violations of MISRA C-2012 rule 17.7.
- 2.3.0
  - Bug Fixes
    - Fixed the build error caused by the SOC has no fifo feature.
- 2.2.3
  - Bug Fixes
    - Corrected the peripheral name in function SAI0_DriverIRQHandler.
- 2.2.2
  - Bug Fixes
    - Fixed the issue of MISRA 2004 rule 9.3.
    - Fixed sign-compare warning.
    - Fixed the PA082 build warning.

        - Fixed sign-compare warning.
        - Fixed violations of MISRA C-2012 rule 10.3,17.7,10.4,8.4,10.7,10.8,14.4,17.7,11.-6,10.1,10.6,8.4,14.3,16.4,18.4.
        - Allow to reset Rx or Tx FIFO pointers only when Rx or Tx is disabled.
    - Improvements
        - Added 24bit raw audio data width support in sai sdma driver.
        - Disabled the interrupt/DMA request in the SAI_Init to avoid generates unexpected sai FIFO requests.
- 2.2.1
    - Improvements
        - Added mclk post divider support in function SAI_SetMasterClockDivider.
        - Removed useless configuration code in SAI_RxSetSerialDataConfig.
    - Bug Fixes
        - Fixed the SAI SDMA driver build issue caused by the wrong structure member name used in the function SAI_TransferRxSetConfigSDMA/SAI_TransferTxSetConfigSDM-A.
        - Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CH-ANNEL_COUNTn.
        - Applied ERR05144: not set FCONT = 1 when TMR $>$ 0, otherwise the TX may not work.
- 2.2.0
    - Improvements
        - Added new APIs for parameters collection and simplified user interfaces:
            - SAI_Init
            - SAI_SetMasterClockConfig
            - SAI_TxSetBitClockRate
            - SAI_TxSetSerialDataConfig
            - SAI_TxSetFrameSyncConfig
            - SAI_TxSetFifoConfig
            - SAI_TxSetBitclockConfig
            - SAI_TxSetConfig
            - SAI_TxSetTransferConfig
            - SAI_RxSetBitClockRate
            - SAI_RxSetSerialDataConfig
            - SAI_RxSetFrameSyncConfig
            - SAI_RxSetFifoConfig
            - SAI_RxSetBitclockConfig
            - SAI_RXSetConfig
            - SAI_RxSetTransferConfig
            - SAI_GetClassicI2SConfig
            - SAI_GetLeftJustifiedConfig
            - SAI_GetRightJustifiedConfig
            - SAI_GetTDMConfig
- 2.1.9
    - Improvements

        * Improved SAI driver comment for clock polarity.
        * Added enumeration for SAI for sample inputs on different edges.
        * Changed FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHANNEL_COUNTn(base) for the difference between the different SAI instances.
    - Added new APIs:
        * SAI_TxSetBitClockDirection
        * SAI_RxSetBitClockDirection
        * SAI_RxSetFrameSyncDirection
        * SAI_TxSetFrameSyncDirection
- 2.1.8
    - Improvements
        * Added feature macro test for the sync mode2 and mode 3.
        * Added feature macro test for masterClockHz in sai_transfer_format_t.
- 2.1.7
    - Improvements
        * Added feature macro test for the mclkSource member in sai_config_t.
        * Changed "FSL_FEATURE_SAI5_SAI6_SHARE_IRQ" to "FSL_FEATURE_SAI_SAI5_SAI6_SHARE_IRQ".
        * Added #ifndef #endif check for SAI_XFER_QUEUE_SIZE to allow redefinition.
    - Bug Fixes
        * Fixed build error caused by feature macro test for mclkSource.
- 2.1.6
    - Improvements
        * Added feature macro test for mclkSourceClockHz check.
        * Added bit clock source name for general devices.
    - Bug Fixes
        * Fixed incorrect channel numbers setting while calling RX/TX set format together.
- 2.1.5
    - Bug Fixes
        * Corrected SAI3 driver IRQ handler name.
        * Added I2S4/5/6 IRQ handler.
        * Added base in handler structure to support different instances sharing one IRQ number.
    - New Features
        * Updated SAI driver for MCR bit MICS.
        * Added 192 KHZ/384 KHZ in the sample rate enumeration.
        * Added multi FIFO interrupt/SDMA transfer support for TX/RX.
        * Added an API to read/write multi FIFO data in a blocking method.
        * Added bclk bypass support when bclk is same with mclk.
- 2.1.4
    - New Features
        * Added an API to enable/disable auto FIFO error recovery in platforms that support this feature.
        * Added an API to set data packing feature in platforms which support this feature.
- 2.1.3
    - New Features

        * Added feature to make I2S frame sync length configurable according to bitWidth.
- 2.1.2
  - Bug Fixes
    * Added 24-bit support for SAI eDMA transfer. All data shall be 32 bits for send/receive, as eDMA cannot directly handle 3-Byte transfer.
- 2.1.1
  - Improvements
    * Reduced code size while not using transactional API.
- 2.1.0
  - Improvements
    * API name changes:
      · SAI_GetSendRemainingBytes -> SAI_GetSentCount.
      · SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount.
      · All names of transactional APIs were added with "Transfer" prefix.
      · All transactional APIs use base and handle as input parameter.
      · Unified the parameter names.
  - Bug Fixes
    * Fixed WLC bug while reading TCSR/RCSR registers.
    * Fixed MOE enable flow issue. Moved MOE enable after MICS settings in SAI_TxInit/-SAI_RxInit.
- 2.0.0
  - Initial version.

SEMA42

The current SEMA42 driver version is 2.0.2.

- 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 10.3, 10.4, 14.4, 18.1.
- 2.0.0
  - Initial version.

TPM

The current TPM driver version is 2.0.7.

- 2.0.7
  - Bug Fixes
    * Fixed violations of MISRA C-2012 rule 10.4 and 17.7.
- 2.0.6

  - Bug Fixes
    - Fixed Out-of-bounds issue.
- 2.0.5
  - Bug Fixes
    - Fixed MISRA-2012 rules.
      - Rule 10.6, 10.7
- 2.0.4
  - Bug Fixes
    - Fixed ERR050050 in functions TPM_SetupPwm/TPM_UpdatePwmDutycycle. When TPM was configured in EPWM mode as PS = 0, the compare event was missed on the first reload/overflow after writing 1 to the CnV register.
- 2.0.3
  - Bug Fixes
    - MISRA-2012 issue fixed.
      - Fixed rules: rule-12.1, rule-17.7, rule-16.3, rule-14.4, rule-1.3, rule-10.4, rule-10.3, rule-10.7, rule-10.1, rule-10.6, and rule-18.1.
- 2.0.2
  - Bug Fixes
    - Fixed issues in functions TPM_SetupPwm/TPM_UpdateChnlEdgeLevelSelect /TPM_SetupInputCapture/TPM_SetupOutputCompare/TPM_SetupDualEdgeCapture, wait acknowledgement when the channel is disabled.
- 2.0.1
  - Bug Fixes
    - Fixed TPM_UpdateChnIEdgeLevelSelect ACK wait issue.
    - Fixed the issue that TPM_SetupdualEdgeCapture could not set FILTER register.
    - Fixed TPM_UpdateChnEdgeLevelSelect ACK wait issue.
- 2.0.0
  - Initial version.

TSTMR

The current TSTMR driver version is 2.0.0.

- 2.0.0
  - Initial version.

WDOG32

The current WDOG32 driver version is 2.0.3.

- 2.0.3
  - Bug Fixes
    - Fixed the noncompliance issue of the reference document.
      - Waited until for new configuration to take effect by checking the RCS bit field.

        - Waited until for registers to be unlocked by checking the ULK bit field.
    - Improvements
        - Added 128 bus clocks delay ensures a smooth transition before restarting the counter with the new configuration when there is no RCS status bit.
- 2.0.2
    - Bug Fixes
        - MISRA C-2012 issue fixed.
            - Fixed rules, containing: rule-10.3, rule-14.4, rule-15.5.
        - Fixed the issue of the inseparable process interrupted by other interrupt source.
            - WDOG32_Refresh
- 2.0.1
    - Bug Fixes
        - WDOG must be configured within its configuration time period.
            - Added WDOG32_Init API to quick access section.
            - Defined register variable in WDOG32_Init API.
- 2.0.0
    - Initial version.

Middleware Change Log

lwIP for MCUXpresso SDK

The current version of lwIP is based on lwIP 2.2.0.dev.

- 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.

Multicore SDK

The current version of Multicore SDK is 2.8.0.

- 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

Component Change Log

SERIAL_MANAGER

The current Serial_Manager component version is 1.0.0.

- 1.0.0
  - Initial version

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.