
MCUXpresso SDK Release Notes Supporting frdmke16z

Change Logs

NXP Semiconductors



Contents

Driver Change Log

ACMP	1
ADC12	1
CLOCK	2
CRC	2
EWM	3
FLASH	3
FTM	4
GPIO	6
LPI2C	7
LPIT	11
LPSPI	11
LPTMR	13
LPUART	13
MMDVSQ	16
MSCAN	17
PDB	18
PMC	18
PORT	19
PWT	19
RCM	19

Title	Page No.
RTC	20
SIM	20
SMC	21
TRGMUX	22
WDOG32	22
CLOCK	22

RTOS Change Log

kernel for MCUXpresso SDK.	24
kernel for MCUXpresso SDK.	26

Middleware Change Log

MOTOR_CONTROL for KSDK	28
RTCESL for KSDK	28
TOUCH_SENSING for KSDK	28

1 Driver Change Log

ACMP

The current ACMP driver version is 2.0.6.

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

ADC12

The current ADC12 driver version is 2.0.6.

- 2.0.6
 - Improvements
 - * Removed useless comments of ADC12_DoAutoCalibration() function.
- 2.0.5
 - Bug Fixes
 - * Fixed the violations of MISRA C-2012 rule 10.4.
- 2.0.4

- Bug Fixes
 - * Fixed the violations of MISRA C-2012 rules:
 - Rule 4.7 10.1 10.3 10.4 10.8 12.2 16.4 17.7
- 2.0.3
 - Improvements
 - * Used conversion control feature macro instead of that in IO map.
- 2.0.2
 - Bug Fixes
 - * Set ADC clock frequency as half of the maximum value for calibration.
- 2.0.1
 - New Features
 - * Added a feature to control enablement of DMA.
- 2.0.0
 - Initial version.

CLOCK

Current CLOCK driver version is 2.3.0

- 2.3.0
 - Bug Fixes
 - * Removed unimplemented kSCG_AsyncDiv1Clk.
- 2.2.1
 - Bug Fixes
 - * Fixed MISRA C-2012 rule 10.1, rule 10.4, rule 10.8 and so on.
- 2.2.0
 - New Features
 - * Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.1.0
 - New Features
 - * Added new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.0.0
 - Initial version.

CRC

The current CRC driver version is 2.0.3.

- 2.0.3
 - Bug fix:
 - * Fix MISRA issues.
- 2.0.2
 - Bug fix:

- * Fix MISRA issues.
- 2.0.1
 - Bug fix:
 - * DATA and DATALL macro definition moved from header file to source file.
- 2.0.0
 - Initial version.

EWM

The current EWM driver version is 2.0.3.

- 2.0.3
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 rules: 10.1, 10.3.
- 2.0.2
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 rules: 10.3, 10.4.
- 2.0.1
 - Bug Fixes
 - * Fixed the hard fault in EWM_Deinit.
- 2.0.0
 - Initial version.

FLASH

Current FLASH driver version is 3.1.2

- 3.1.2
 - Bug Fixes — Remove redundant comments.
- 3.1.1
 - Bug Fixes — MISRA C-2012 issue fixed: rule 10.3
- 3.1.0
 - New Feature
 - * Support erase flash asynchronously.
- 3.0.2
 - Bug Fixes — MISRA C-2012 issue fixed: rule 8.4, 17.7, 10.4, 16.1, 21.15, 11.3, 10.7 — building warning -Wnull-dereference on arm compiler v6
- 3.0.1
 - New Features
 - * Added support FlexNVM alias for (kw37/38/39).
- 3.0.0
 - Improvements
 - * Reorganized FTFx flash driver source file.
 - * Extracted flash cache driver from FTFx driver.

- * Extracted flexnvm flash driver from FTFx driver.
- 2.3.1
 - Bug Fixes
 - * Unified Flash IFR design from K3.
 - * New encoding rule for K3 flash size.
- 2.3.0
 - New Features
 - * Added support for device with LP flash (K3S/G).
 - * Added flash prefetch speculation APIs.
 - Improvements
 - * Refined flash_cache_clear function.
 - * Reorganized the member of flash_config_t struct.
- 2.2.0
 - New Features
 - * Supported FTL device in FLASH_Swap API.
 - * Supported various pflash start addresses.
 - * Added support for KV58 in cache clear function.
 - * Added support for device with secondary flash (KW40).
 - Bug Fixes
 - * Compiled execute-in-ram functions as PIC binary code for driver use.
 - * Added missed flexram properties.
 - * Fixed unaligned variable issue for execute-in-ram function code array.
- 2.1.0
 - Improvements
 - * Updated coding style to align with KSDK 2.0.
 - * Different-alignment-size support for pflash and flexnvm.
 - * Improved the implementation of execute-in-ram functions.
- 2.0.0
 - Initial version

FTM

The current FTM driver version is 2.5.0.

- 2.5.0
 - Improvements
 - * Added FTM_CalculateCounterClkDiv to help calculates the counter clock prescaler.
 - * Modify FTM_UpdatePwmDutycycle API to make it return pwm duty cycles status.
 - Bug Fixes
 - * Fixed TPM_SetupPwm can't configure 100% center align combined PWM issues.
- 2.4.1
 - Bug Fixes
 - * Added function macro to determine if FTM instance has only basic features, to prevent access to protected register bits.

- 2.4.0
 - Improvements
 - * Added CNTIN register initialization in FTM_SetTimerPeriod API.
 - * Added a new API to read the captured value of a FTM channel configured in capture mode:
 - FTM_GetInputCaptureValue()
- 2.3.0
 - Improvements
 - * Added support of EdgeAligned/CenterAligned/Asymmetrical combine PWM mode in FTM_SetupPWM() and FTM_SetupPwmMode() APIs.
 - * Remove kFTM_ComplementaryPwm from support PWM mode, and add new parameter "enableComplementary" in structure ftm_chnl_pwm_signal_param_t.
 - * Rename FTM_SetupFault() API to FTM_SetupFaultInput() to avoid ambiguity.
- 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.

GPIO

The current driver version is 2.6.0.

- 2.6.0
 - New Features
 - * Added API to get GPIO version information.
 - * Added API to control a pin for general purpose input.
 - * Added some APIs to control pin in secure and privilege status.
- 2.5.3
 - Bug Fixes
 - * Correct the feature macro typo: FSL_FEATURE_GPIO_HAS_NO_INDEP_OUTPUT_-CONTORL.
- 2.5.2
 - Improvements
 - * Improved GPIO_PortSet/GPIO_PortClear/GPIO_PortToggle functions to support devices without Set/Clear/Toggle registers.
- 2.5.1
 - Bug Fixes
 - * Fixed wrong macro definition.
 - * Fixed MISRA C-2012 rule issues in the FGPIO_CheckAttributeBytes() function.
 - * Defined the new macro to separate the scene when the width of registers is different.
 - * Removed some redundant macros.
 - New Features
 - * Added some APIs to get/clear the interrupt status flag when the port doesn't control pins'

interrupt.

- 2.4.1
 - Improvements
 - * Improved GPIO_CheckAttributeBytes() function to support 8 bits width GACR register.
- 2.4.0
 - Improvements
 - * API interface added:
 - New APIs were added to configure the GPIO interrupt clear settings.
- 2.3.2
 - Bug Fixes
 - * Fixed the issue for MISRA-2012 check.
 - Fixed rule 3.1, 10.1, 8.6, 10.6, and 10.3.
- 2.3.1
 - Improvements
 - * Removed deprecated APIs.
- 2.3.0
 - New Features
 - * Updated the driver code to adapt the case of interrupt configurations in GPIO module. New APIs were added to configure the GPIO interrupt settings if the module has this feature on it.
- 2.2.1
 - Improvements
 - * API interface changes:
 - Refined naming of APIs while keeping all original APIs by marking them as deprecated. The original APIs will be removed in next release. The main change is updating APIs with prefix of _PinXXX() and _PortXXX.
- 2.1.1
 - Improvements
 - * API interface changes:
 - Added an API for the check attribute bytes.
- 2.1.0
 - Improvements
 - * API interface changes:
 - Added "pins" or "pin" to some APIs' names.
 - Renamed "_PinConfigure" to "GPIO_PinInit".

LPI2C

The current LPI2C driver version is 2.3.1.

- 2.3.1
 - Improvements
 - * Updated LPI2C_GetCyclesForWidth to add the parameter of minimum cycle, because for master SDA/SCL filter, master bus idle/pin low timeout and slave SDA/SCL filter

configuration, 0 means disabling the feature and cannot be used.

– Bug Fixes

- * Fixed bug in LPI2C_SlaveTransferHandleIRQ that when restart detect event happens the transfer structure should not be cleared.
- * Fixed bug in LPI2C_RunTransferStateMachine, that when only slave address is transferred or there is still data remaining in tx FIFO the last byte's nack cannot be ignored.
- * Fixed bug in slave filter doze enable, that when FILTDZ is set it means disable rather than enable.
- * Fixed bug in the usage of LPI2C_GetCyclesForWidth. First its return value cannot be used directly to configure the slave FILTSDA, FILTSCL, DATAVD or CLKHOLD, because the real cycle width for them should be FILTSDA+3, FILTSCL+3, FILTSCL+DATAVD+3 and CLKHOLD+3. Second when cycle period is not affected by the prescaler value, prescaler value should be passed as 0 rather than 1.
- * Fixed wrong default setting for LPI2C slave. If enabling the slave tx SCL stall, then the default clock hold time should be set to 250ns according to I2C spec for 100kHz standard mode baudrate.
- * Fixed bug that before pushing command to the tx FIFO the FIFO occupation should be checked first in case FIFO overflow.

• 2.3.0

– New Features

- * Supported reading more than 256 bytes of data in one transfer as master.
- * Added API LPI2C_GetInstance.

– Bug Fixes

- * Fixed bug in LPI2C_MasterTransferAbortEDMA, LPI2C_MasterTransferAbort and LPI2C_MasterTransferHandleIRQ that before sending stop signal whether master is active and whether stop signal has been sent should be checked, to make sure no FIFO error or bus error will be caused.
- * Fixed bug in LPI2C master EDMA transactional layer that the bus error cannot be caught and returned by user callback, by monitoring bus error events in interrupt handler.
- * Fixed bug in LPI2C_GetCyclesForWidth that the parameter used to calculate clock cycle should be $2^{\text{prescaler}}$ rather than prescaler.
- * Fixed bug in LPI2C_MasterInit that timeout value should be configured after baudrate, since the timeout calculation needs prescaler as parameter which is changed during baudrate configuration.
- * Fixed bug in LPI2C_MasterTransferHandleIRQ and LPI2C_RunTransferStateMachine that when master writes with no stop signal, need to first make sure no data remains in the tx FIFO before finishes the transfer.

• 2.2.0

– Bug Fixes

- * Fixed issue that the SCL high time, start hold time and stop setup time do not meet I2C specification, by changing the configuration of data valid delay, setup hold delay, clock high and low parameters.
- * MISRA C-2012 issue fixed.
 - Fixed rule 8.4, 13.5, 17.7, 20.8.

- 2.1.12
 - Bug Fixes
 - * Fixed MISRA advisory 15.5 issues.
- 2.1.11
 - Bug Fixes
 - * Fixed the bug that, during master non-blocking transfer, after the last byte is sent/received, the kLPI2C_MasterNackDetectFlag is expected, so master should not check and clear kLPI2C_MasterNackDetectFlag when remainingBytes is zero, in case FIFO is emptied when stop command has not been sent yet.
 - * Fixed the bug that, during non-blocking transfer slave may nack master while master is busy filling tx FIFO, and NDF may not be handled properly.
- 2.1.10
 - Bug Fixes
 - * MISRA C-2012 issue fixed.
 - Fixed rule 10.3, 14.4, 15.5.
 - * Fixed unaligned access issue in LPI2C_RunTransferStateMachine.
 - * Fixed uninitialized variable issue in LPI2C_MasterTransferHandleIRQ.
 - * Used linked TCD to disable tx and enable rx in read operation to fix the issue that for platform sharing the same DMA request with tx and rx, during LPI2C read operation if interrupt with higher priority happened exactly after command was sent and before tx disabled, potentially both tx and rx could trigger dma and cause trouble.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 11.6, 11.9, 14.4, 17.7.
 - * Fixed the waitTimes variable not re-assignment issue for each byte read.
 - New Features
 - * Added the IRQHandler for LPI2C5 and LPI2C6 instances.
 - Improvements
 - * Updated the LPI2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
- 2.1.9
 - Bug Fixes
 - * Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
 - * Fixed Coverity issue of operands did not affect the result in LPI2C_SlaveReceive and LPI2C_SlaveSend.
 - * Removed STOP signal wait when NAK detected.
 - * Cleared slave repeat start flag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved repeat start flag. This caused the next slave to send a break, and the master was always in the receive data status, but could not receive data.
- 2.1.8
 - Bug Fixes
 - * Fixed the transfer issue with LPI2C_MasterTransferNonBlocking, kLPI2C_TransferNoStopFlag, with the wait transfer done through callback in a way of not doing a blocking transfer.
 - * Fixed the issue that STOP signal did not appear in the bus when NAK event occurred.
- 2.1.7

- Bug Fixes
 - * Cleared the stopflag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved stop flag and caused the next slave to send a break, and the master always stayed in the receive data status but could not receive data.
- 2.1.6
 - Bug Fixes
 - * Fixed driver MISRA build error and C++ build error in LPI2C_MasterSend and LPI2C_SlaveSend.
 - * Reset FIFO in LPI2C Master Transfer functions to avoid any byte still remaining in FIFO during last transfer.
 - * Fixed the issue that LPI2C_MasterStop did not return the correct NAK status in the bus for second transfer to the non-existing slave address.
- 2.1.5
 - Bug Fixes
 - * Extended the Driver IRQ handler to support LPI2C4.
 - * Changed to use ARRAY_SIZE(kLpi2cBases) instead of FEATURE COUNT to decide the array size for handle pointer array.
- 2.1.4
 - Bug Fixes
 - * Fixed the LPI2C_MasterTransferEDMA receive issue when LPI2C shared same request source with TX/RX DMA request. Previously, the API used scatter-gather method, which handled the command transfer first, then the linked TCD which was pre-set with the receive data transfer. The issue was that the TX DMA request and the RX DMA request were both enabled, so when the DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happened due to empty TX FIFO. The result was that the RX DMA transfer would start without waiting on the expected RX DMA request.
 - * Fixed the issue by enabling IntMajor interrupt for the command TCD and checking if there was a linked TCD to disable the TX DMA request in LPI2C_MasterEDMACallback API.
- 2.1.3
 - Improvements
 - * Added LPI2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
 - * Added LPI2C_MasterTransferBlocking API.
- 2.1.2
 - Bug Fixes
 - * In LPI2C_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag was detected.
- 2.1.1
 - Bug Fixes
 - * Disabled the auto-stop feature in eDMA driver. Previously, the auto-stop feature was enabled at transfer when transferring with stop flag. Since transfer was without stop flag and the auto-stop feature was enabled, when starting a new transfer with stop flag, the stop

flag would be sent before the new transfer started, causing unsuccessful 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 2.0.2.

- 2.0.2
 - Improvements
 - * Improved LPIT_SetTimerPeriod implementation, configure timeout value with LPIT ticks minus 1 generate more correct interval.
 - * Added timeout value configuration check for LPIT_SetTimerPeriod, at least input 3 ticks for calling LPIT_SetTimerPeriod.
 - Bug Fixes
 - * Fixed MISRA C-2012 rule 17.7 violations.
- 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 2.2.1.

- 2.2.1
 - Bug Fixes
 - * Fixed bug in LPSPi_SetPCSContinuous when disabling PCS continuous mode.
- 2.2.0

- Bug Fixes
 - * Fixed bug in 3-wire polling and interrupt transfer that the received data is not correct and the PCS continuous mode is not working.
- 2.1.0
 - Improvements
 - * Improved LPSPI_SlaveTransferHandleIRQ to fill up TX FIFO instead of write one data to TX register which improves the slave transmit performance.
 - * Added new functional APIs LPSPI_SelectTransferPCS and LPSPI_SetPCSContinuous to support changing PCS selection and PCS continuous mode.
 - Bug Fixes
 - * Fixed bug in non-blocking and EDMA transfer APIs that kStatus_InvalidArgument is returned if user configures 3-wire mode and full-duplex transfer at the same time, but transfer state is already set to kLPSPI_Busy by mistake causing following transfer can not start.
 - * Fixed bug when LPSPI slave using EDMA way to transfer, tx should be masked when tx data is null, otherwise in 3-wire mode which tx/rx use the same pin, the received data will be interfered.
- 2.0.5
 - Improvements
 - * Added timeout mechanism when waiting certain states in transfer driver.
 - Bug Fixes
 - * Fixed the bug that LPSPI can not transfer large data using EDMA.
 - * Fixed MISRA 17.7 issues.
 - * Fixed variable overflow issue introduced by MISRA fix.
 - * Fixed issue that rxFifoMaxBytes should be calculated according to transfer width rather than FIFO width.
 - * Fixed issue that completion flag was not cleared after transfer completed.
- 2.0.4
 - Bug Fixes
 - * Fixed in LPSPI_MasterTransferBlocking that master rxfifo may overflow in stall condition.
 - * Eliminated IAR Pa082 warnings.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 10.6, 11.9, 14.2, 14.4, 15.7, 17.7.
- 2.0.3
 - Bug Fixes
 - * Removed LPSPI_Reset from LPSPI_MasterInit and LPSPI_SlaveInit, because this API may glitch the slave select line. If needed, call this function manually.
- 2.0.2
 - New Features
 - * Added dummy data set up API to allow users to configure the dummy data to be transferred.
 - * Enabled the 3-wire mode, SIN and SOUT pins can be configured as input/output pin.
- 2.0.1
 - Bug Fixes

- * Fixed the bug that the clock source should be divided by the PRESCALE setting in LPSPI_MasterSetDelayTimes function.
 - * Fixed the bug that LPSPI_MasterTransferBlocking function would hang in some corner cases.
- Optimization
 - * Added #ifndef/#endif to allow user to change the default TX value at compile time.
- 2.0.0
 - Initial version.

LPTMR

The current LPTMR driver version is 2.1.1.

- 2.1.1
 - Improvements
 - * Updated the characters from "PTMR" to "LPTMR" in "FSL_FEATURE_PTMR_HAS_NO_PRESCALER_CLOCK_SOURCE_1_SUPPORT" feature definition.
- 2.1.0
 - Improvements
 - * Implement for some special devices' not supporting for all clock sources.
 - Bug Fixes
 - * Fixed issue when accessing CMR register.
- 2.0.2
 - Bug Fixes
 - * Fixed MISRA-2012 issues.
 - Rule 10.1.
- 2.0.1
 - Improvements
 - * Updated the LPTMR driver to support 32-bit CNR and CMR registers in some devices.
- 2.0.0
 - Initial version.

LPUART

The current LPUART driver version is 2.5.3.

- 2.5.3
 - Bug Fixes
 - * Fixed comments by replacing unused status flags kLPUART_NoiseErrorInRxDataRegFlag and kLPUART_ParityErrorInRxDataRegFlag with kLPUART_NoiseErrorFlag and kLPUART_ParityErrorFlag.
- 2.5.2
 - Bug Fixes
 - * Fixed bug that when setting watermark for TX or RX FIFO, the value may exceed the

- maximum limit.
 - Improvements
 - * Added check in LPUART_TransferDMAHandleIRQ and LPUART_TransferEdmaHandleIRQ to ensure if user enables any interrupts other than transfer complete interrupt, the dma transfer is not terminated by mistake.
- 2.5.1
 - Improvements
 - * Use separate data for TX and RX in lpuart_transfer_t.
 - Bug Fixes
 - * Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling LPUART_TransferReceiveNonBlocking, the received data count returned by LPUART_TransferGetReceiveCount is wrong.
- 2.5.0
 - Bug Fixes
 - * Added missing interrupt enable masks kLPUART_Match1InterruptEnable and kLPUART_Match2InterruptEnable.
 - * Fixed bug in LPUART_EnableInterrupts, LPUART_DisableInterrupts and LPUART_GetEnabledInterrupts that the BAUD[LBKDIE] bit field should be soc specific.
 - * Fixed bug in LPUART_TransferHandleIRQ that idle line interrupt should be disabled when rx data size is zero.
 - * Deleted unused status flags kLPUART_NoiseErrorInRxDataRegFlag and kLPUART_ParityErrorInRxDataRegFlag, since firstly their function are the same as kLPUART_NoiseErrorFlag and kLPUART_ParityErrorFlag, secondly to obtain them one data word must be read out thus interfering with the receiving process.
 - * Fixed bug in LPUART_GetStatusFlags that the STAT[LBKDIF], STAT[MA1F] and STAT[MA2F] should be soc specific.
 - * Fixed bug in LPUART_ClearStatusFlags that tx/rx FIFO is reset by mistake when clearing flags.
 - * Fixed bug in LPUART_TransferHandleIRQ that while clearing idle line flag the other bits should be masked in case other status bits be cleared by accident.
 - * Fixed bug of race condition during LPUART transfer using transactional APIs, by disabling and re-enabling the global interrupt before and after critical operations on interrupt enable register.
 - * Fixed DMA/eDMA transfer blocking issue by enabling tx idle interrupt after DMA/eDMA transmission finishes.
 - New Features
 - * Added APIs LPUART_GetRxFifoCount/LPUART_GetTxFifoCount to get rx/tx FIFO data count.
 - * Added APIs LPUART_SetRxFifoWatermark/LPUART_SetTxFifoWatermark to set rx/tx FIFO water mark.
- 2.4.1
 - Bug Fixes
 - * Fixed MISRA advisory 17.7 issues.
- 2.4.0
 - New Features

- * Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.1
 - Bug Fixes
 - * Fixed MISRA advisory 15.5 issues.
- 2.3.0
 - Improvements
 - * Modified LPUART_TransferHandleIRQ so that txState will be set to idle only when all data has been sent out to bus.
 - * Modified LPUART_TransferGetSendCount so that this API returns the real byte count that LPUART has sent out rather than the software buffer status.
 - * Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.8
 - Bug Fixes
 - * Fixed issue for MISRA-2012 check.
 - Fixed rule-10.3, rule-14.4, rule-15.5.
 - * Eliminated Pa082 warnings by assigning volatile variables to local variables and using local variables instead.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 10.8, 14.4, 11.6, 17.7.
 - Improvements
 - * Added check for kLPUART_TransmissionCompleteFlag in LPUART_WriteBlocking, LPUART_TransferHandleIRQ, LPUART_TransferSendDMACallback and LPUART_SendEDMACallback to ensure all the data would be sent out to bus.
 - * Rounded up the calculated sbr value in LPUART_SetBaudRate and LPUART_Init to achieve more accurate baudrate setting. Changed osr from uint32_t to uint8_t since osr's biggest 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.

MMDVVSQ

The current MMDVVSQ driver version is 2.0.3.

- 2.0.3
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 10.3, 10.4, and 14.4.
- 2.0.2
 - Bug fix:
 - * Fixed `MMDVVSQ_GetExecutionStatus` function get execution status wrong.

- 2.0.1
 - Other changes:
 - * Changed name of MMDVVSQ_GetDivideRemainder and MMDVVSQ_GetDivideQuotient functions.
- 2.0.0
 - Initial version.

MSCAN

The current MSCAN driver version is 2.0.7.

- 2.0.7
 - Bug Fixes
 - * Fix build warning.
- 2.0.6
 - Bug Fixes
 - * Fix MISRA C-2012 rule 10.3.
- 2.0.5
 - Bug Fixes
 - * Fixed the code error issue and simplified the algorithm in improved timing APIs.
 - MSCAN_CalculateImprovedTimingValues
- 2.0.4
 - Bug Fixes
 - * Fixed the non-divisible case in improved timing API.
 - MSCAN_CalculateImprovedTimingValues
- 2.0.3
 - Bug Fixes
 - * MISRA C-2012 issue check.
 - Fixed MISRA C-2012 issue in mcan driver, containing: rule-10.1, rule-10.3, rule-10.4, rule-10.7, rule-11.8, rule-14.4, rule-15.5, rule-16.1, rule-16.4, rule-17.7, rule-8.4, rule-8.5.
- 2.0.2
 - Bug Fixes
 - * Fixed issue Clang - disable declaration alignment.
 - * Fixed issue Central repository code formatting.
 - Improvements
 - * Implemented feature to find classical CAN improved timing configuration.
- 2.0.1
 - Bug Fixes
 - * Fixed timestamp issue where it cannot be enabled.
 - * Fixed standard mode frame buffer configuration.
 - * Added RX Message Buffer Mask helper macro, MSCAN_RX_MB_STD_MASK, MSCAN_RX_MB_EXT_MASK.
 - * Fixed dataByte[0-7] order in struct type mscan_frame_t.

- * Updated function MSCAN_WriteTxMb. The MSCAN registers are 8 bits, using 8 bits write for registers.
- 2.0.0
 - Initial version.

PDB

The current PDB driver version is 2.0.4.

- 2.0.4
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.1 and 10.4.
- 2.0.3
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 17.7.
- 2.0.2
 - Improvement:
 - * Used macros in feature file instead of that in IO map.
- 2.0.1
 - Changed PDB register base array to const.
- 2.0.0
 - Initial version.

PMC

The current PMC driver version is 2.0.3.

- 2.0.3
 - Bug Fixes
 - * Fixed the violation of MISRA C-2012 rule 11.3.
- 2.0.2
 - Bug Fixes
 - * Fixed the violations of MISRA 2012 rules:
 - Rule 10.3.
- 2.0.1
 - Bug Fixes
 - * Fixed MISRA issues.
 - Rule 10.8, Rule 10.3.
- 2.0.0
 - Initial version.

PORT

The current PORT driver version is 2.3.0.

- 2.3.0
 - New Features
 - * Added new APIs for Electrical Fast Transient(EFT) detect.
 - * Added new API to configure port voltage range.
- 2.2.0
 - New Features
 - * Added new api PORT_EnablePinDoubleDriveStrength.
- 2.1.1
 - Bug Fixes
 - * Fixed the violations of MISRA C-2012 rules: 10.1, 10.411.311.8, 14.4.
- 2.1.0
 - New Features
 - * Updated the driver code to adapt the case of the interrupt configurations in GPIO module.
 - Will move the pin configuration APIs to GPIO module.
- 2.0.2
 - Other Changes
 - * Added feature guard macros in the driver.
- 2.0.1
 - Other Changes
 - * Added "const" in function parameter.
 - * Updated some enumeration variables' names.

PWT

The current PWT driver version is 2.0.1.

- 2.0.1
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rules: 10.8, 10.3, 10.6.
- 2.0.0
 - Initial version.

RCM

The current RCM driver version is 2.0.4.

- 2.0.4
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 rule 10.3
- 2.0.3
 - Bug Fixes

- * Fixed violation of MISRA C-2012 rules.
- 2.0.2
 - Bug Fixes
 - * Fixed MISRA issue.
 - Rule 10.8, rule 10.1, rule 13.2, rule 3.1.
- 2.0.1
 - Bug Fixes
 - * Fixed kRCM_SourceSw bit shift issue.
- 2.0.0
 - Initial version.

RTC

The current RTC driver version is 2.2.1.

- 2.2.1
 - Bug Fixes
 - * Fixed the issue of Pa082 warning.
 - * Fixed the issue of bit field mask checking.
 - * Fixed the issue of hard code in RTC_Init.
- 2.2.0
 - Bug Fixes
 - * Fixed MISRA C-2012 issue.
 - Fixed rule contain: rule-17.7, rule-14.4, rule-10.4, rule-10.7, rule-10.1, rule-10.3.
 - * Fixed central repository code formatting issue.
 - Improvements
 - * Added an API for enabling wakeup pin.
- 2.1.0
 - Improvements
 - * Added feature macro check for many features.
- 2.0.0
 - Initial version.

SIM

The current SIM driver version is 2.1.3.

- 2.1.3
 - Improvements
 - * Updated function SIM_GetUniqueId to support different register names.
- 2.1.2
 - Bug Fixes
 - * Fixed SIM_GetUniqueId bug that could not get UIDH.
- 2.1.1

- Bug Fixes
 - * Fixed violations of the MISRA C-2012 rules 10.1, 10.4
- 2.1.0
 - Improvements
 - * Added new APIs: SIM_GetRfAddr() and SIM_EnableSysTickClock().
- 2.0.0
 - Initial version.

SMC

The current SMC driver version is 2.0.7.

- 2.0.7
 - Bug Fixes
 - * Fixed MISRA-2012 issue 10.3.
- 2.0.6
 - Bug Fixes
 - * Fixed issue for MISRA-2012 check.
 - Fixed rule 10.3, rule 11.3.
- 2.0.5
 - Bug Fixes
 - * Fixed issue for MISRA-2012 check.
 - Fixed rule 15.7, rule 14.4, rule 10.3, rule 10.1, rule 10.4.
- 2.0.4
 - Bug Fixes
 - * When entering stop modes, used RAM function for the flash synchronization issue. Application should make sure that, the RW data of fsl_smc.c is located in memory region which is not powered off in stop modes.
- 2.0.3
 - Improvements
 - * Added APIs SMC_PreEnterStopModes, SMC_PreEnterWaitModes, SMC_PostExitWaitModes, and SMC_PostExitStopModes.
- 2.0.2
 - Bug Fixes
 - * Added DSB before WFI while ISB after WFI.
 - Other Changes
 - * Updated SMC_SetPowerModeVlpw implementation.
- 2.0.1
 - Other Changes
 - * Updated for KL8x.
- 2.0.0
 - Initial version.

TRGMUX

The current TRGMUX driver version is 2.0.1.

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

WDOG32

The current WDOG32 driver version is 2.0.4.

- 2.0.4
 - Improvements
 - * To ensure that the reconfiguration is inside 128 bus clocks unlock window, put all re-configuration APIs in quick access code section.
- 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.

CLOCK

Current CLOCK driver version is 2.0.0

- 2.0.0

- Initial version.

2 RTOS Change Log

kernel for MCUXpresso SDK.

The current version is Amazon-FreeRTOS 202107.00 Original package is available at github.com/aws/amazon-freertos.

- 202107.00_rev1
 - Apply CM33 kernel security fix from 10.4.3-LTS-Patch-2. See: rtos\freertos\freertos_kernel\History.txt https://www.freertos.org/security/security_updates.html <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-43997>
 - Apply CM33 kernel security fix from 10.4.3-LTS-Patch-1. See rtos\freertos\freertos_kernel\History.txt
- 202107.00_rev0
 - update amazon freertos version.
- 202012.00_rev0
 - update amazon freertos version.
 - update libraries/logging from github.com/aws/amazon-freertos (commit fc1d0e3c72ca791624477474bc7b)
 - increase mqttexamplePROCESS_LOOP_TIMEOUT_MS to fix the assert error that is caused by time out.
- 202007.00_rev0
 - update amazon freertos version.
- 202002.00_rev2
 - updated aws_iot_ota_agent.c for enabling OTA canceling (new state transition from eOTA_AgentState_WaitingForFileBlock to eOTA_AgentEvent_ReceivedJobDocument)
- 202002.00_rev1
 - updated iot_tls.c as per latest SSS stack v2.14. All SSS/SE05x code under conditional compilation.
 - fix "#pragma weak" issue caused by IAR update to version 8.50.5. Use __weak definition of vPortSetupTimerInterrupt instead of "#pragma weak" declaration.
- 202002.00_rev0
 - update aws_iot_network_manager
 - enable lowpower tickless for CA9, CM33
- 201908.00_rev0
 - update amazon freertos version
 - Fix freertos_tasks_c_additions.h - fix IAR build fail
 - update queue.c - add definition for pvBuffer necessary for segger sysview
 - iot_crypto.c - change include file to be possible include mbedtls config file defined by MBED-TLS_CONFIG_FILE macro
 - iot_mqtt_agent.h - extend MQTTAgentConnectParams_t structure - required by se_hostlib examples
 - Fixed build warnings:
 - * aws_dev_mode_key_provisioning.c - some variables were declared but never referenced
 - * aws_iot_network_manager.c - some functions were declared but never referenced

- * `iot_device_metrics.c` - add include
- * `iot_pkcs11_mbedtls.c` - incompatible pointer type, unused variable
- * `iot_demo_freertos.c` - macro expansion producing 'defined' has undefined behavior
- * `iot_pkcs11_mbedtls.c` - comparison of address not equal to a null pointer is always true
- * `pkcs11.h` - '__PASTE' macro redefined
- changes required for `se_hostlib` examples
 - * `iot_default_root_certificates.h` `tlsCombi_ROOT_CERTIFICATE_PEM` added for `se_hostlib` cloud demos
 - * `iot_mqtt_agent.h` modified structure to support for clouds which use username and password
 - * `iot_mqtt_agent.c` assigning structure to support for clouds which use username and password
 - * `iot_crypto.c` threading alt under `MBEDTLS_THREADING_ALT`
 - * `iot_tls.c` Support for secure element handling
 - * `aws_pkcs11_pal.c` `pkcs11` handling for SE050
- Introduced `aws_ota_pal.c` for RT1060 supporting `ota_bootloader` (mcu-boot)
- updated lwIP template with options for core locking
- 1.4.9_rev0
 - Remove 3rd party libraries lwip, mbedtls (use MCUXpresso SDK versions).
 - Add missing comments to `heap_useNewlib.c`.
- 1.4.7_rev0
 - New features:
 - * Add optional allocation scheme `heap_useNewlib.c` by D. Nadler.
 - * Enable task aware debugging for cm33 platforms
 - * Move tickless implementation to application layer
 - Other changes:
 - * Fix other build warnings, errors
- 1.4.6_rev0
 - New features:
 - * Update support of CM33 port with Trustzone, MPU, FPU support
 - * Add support for AWS test for Cypress WiFi
 - * Use lwip netif api to avoid lwIP raw API calls outside of tcpip thread in `aws_wifi.c`
 - Other changes:
 - * Fix issues with mflash driver
 - * Fix other build warnings, errors
- 1.4.0_rev1
 - New features:
 - * Add implementation of `vTaskEndScheduler` for CM0 GCC port.
 - * Support for CM33, CM33F architectures based on CM3, CM4F ports
- 1.4.0_rev0
 - New features:
 - * Support for `pkcs11` for several platforms, secure element host library under `pkcs11/portable/nxp` folder
 - * Lwip, `wifi_qca` support for `secure_sockets` in `secure_sockets/portable/nxp` folder
 - * Flash driver support for several platforms in `third_party/mcu_vendor/nxp` folder

- * Generic support for aws_wifi under wifi/portable/nxp/common folder
- Other changes:
 - * Fix several build warnings, errors

kernel for MCUXpresso SDK.

The current version is FreeRTOS kernel 10.4.3-LTS-Patch-2 Original package is available at github.com/FreeRTOS/FreeRTOS-Kernel.

- 10.4.3_rev1
 - Apply CM33 security fix from 10.4.3-LTS-Patch-2. See rtos\freertos\freertos_kernel\History.txt
 - Apply CM33 security fix from 10.4.3-LTS-Patch-1. See rtos\freertos\freertos_kernel\History.txt
- 10.4.3_rev0
 - update amazon freertos version.
- 10.4.3_rev0
 - update amazon freertos version.
- 9.0.0_rev3
 - New features:
 - * Tickless idle mode support for Cortex-A7. Add fsl_tickless_epit.c and fsl_tickless_generic.h in portable/IAR/ARM_CA9 folder.
 - * Enabled float context saving in IAR for Cortex-A7. Added configUSE_TASK_FPU_SUPPORT macros. Modified port.c and portmacro.h in portable/IAR/ARM_CA9 folder.
 - Other changes:
 - * Transformed ARM_CM core specific tickless low power support into generic form under freertos/Source/portable/low_power_tickless/.
- 9.0.0_rev2
 - New features:
 - * Enabled MCUXpresso thread aware debugging. Add freertos_tasks_c_additions.h and configINCLUDE_FREERTOS_TASK_C_ADDITIONS_H and configFRTOS_MEMORY_SCHEME macros.
- 9.0.0_rev1
 - New features:
 - * Enabled -fno-plt optimization in GCC by adding **attribute((used))** for vTaskSwitchContext.
 - * Enabled KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD_STACK_HIGH_ADDRESS macro. Modified files are task.c and FreeRTOS.h.
- 9.0.0_rev0
 - New features:
 - * Example freertos_sem_static.
 - * Static allocation support RTOS driver wrappers.
 - Other changes:
 - * Tickless idle rework. Support for different timers is in separated files (fsl_tickless_systick.c, fsl_tickless_lptmr.c).

- * Removed configuration option configSYSTICK_USE_LOW_POWER_TIMER. Low power timer is now selected by linking of appropriate file fsl_tickless_lptmr.c.
- * Removed configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of **attribute((weak))** is the preferred solution. Not same as _weak!
- 8.2.3
 - New features:
 - * Tickless idle mode support.
 - * Added template application for Kinetis Expert (KEx) tool (template_application).
 - Other changes:
 - * Folder structure reduction. Keep only Kinetis related parts.

3 Middleware Change Log

MOTOR_CONTROL for KSDK

Current version is 1.1.0

- 1.1.0
 - Initial version.

RTCESL for KSDK

Current version is 4.3

- 4.3
 - Initial version.

TOUCH_SENSING for KSDK

Current version is 1.1.0

- 1.1.0
 - 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.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, the Freescale logo, Kinetis, Processor Expert, and Tower are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex, Keil, Mbed, Mbed Enabled, and Vision are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2021 NXP B.V.

