
MCUXpresso SDK Release Notes Supporting frdmke04z

Change Logs

NXP Semiconductors



Contents

Driver Change Log

PORT	1
CLOCK	1
I2C	1
SPI_CMSIS	2
UART	2
ACMP	3
ADC	3
COMMON	3
FLASH	5
FTM	6
GPIO	7
I2C	8
IRQ	10
KBI	10
PIT	10
PORT	11
PWT	11
RTC	12
SPI	12
TPM	13

Title	Page No.
UART	15
WDOG8	17

1 Driver Change Log

PORT

The current PORT driver version is 2.0.2.

- 2.0.2
 - Bug Fixes
 - * Fixed issue for MISRA-2012 check.
 - Fixed rule 10.1, rule 10.4, rule 10.7.
- 2.0.1
 - Change pin index enum port_pin_index_t to uint8_t in PORT_SetPinPullUpEnable();
- 2.0.0
 - initial version;

CLOCK

The current CLOCK driver version is 2.2.1.

- 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 feature:
 - * Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.1.0
 - New feature
 - * Adding new API CLOCK_DelayAtLeastUs() to implement a delay function which allows users set delay in unit of microsecond.
- 2.0.3
 - some minor fixes.
- 2.0.0
 - initial version.

I2C

Current I2C CMSIS driver version is 2.2.0

- 2.2.0
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.0.1

- Bug Fixes
 - * In ARM_I2C_ABORT_TRANSFER operation in I2C_InterruptControl, the method to check if I2C is operating as slave is not correct, then master may have potential risk to block at the slave check code.
- 2.0.0
 - Initial version.

SPI_CMSIS

Current spi_cmsis driver version is 2.3

- 2.3
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 10.8, 11.8, 11.9, 14.4, 16.1, 16.3, 16.4, 17.7, 20.9.
- 2.2
 - Bug Fixes
 - * Fixed the bug that, the parameter num of APIs ARM_SPI_Transfer, ARM_SPI_Send and ARM_SPI_Receive, and the return value of API ARM_SPI_GetDataCount should be the number of data item defined by datawidth, rather than the number of byte.
- 2.1
 - Bug Fixes
 - * Fixed the wrong clock polarity assignment in driver. For ARM_SPI_CPOL0_CPHA0 and other frame format parameters, CPOL = 0 means kSPI_ClockPolarityActiveHigh not kSPI_ClockPolarityActiveLow in driver.
 - New features
 - * Added new feature to support 3-wire mode for both master and slave instance. User can use ARM_SPI_MODE_MASTER_SIMPLEX to set the MOSI pin as output/input pin, and use the ARM_SPI_MODE_SLAVE_SIMPLEX to set the MISO pin as output/input pin.
 - * Allowed user to set up the default transmit value by using ARM_SPI_SET_DEFAULT_TX_VALUE.
- 2.0
 - Initial version.

UART

The current UART CMSIS driver version is 2.1

- 2.1
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.48.610.110.310.411.111.914.415.716.116.316.416.617.720.720.9.
- 2.0

- Initial version.

ACMP

The current ACMP driver version is 2.0.2.

- 2.0.2
 - 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 violation of MISRA C-2012 rules:
 - Rule 3.1 8.3 10.3 17.7.
- 2.0.1
 - Bug Fixes
 - * Fixed the missing right pair definition for extern C.
- 2.0.0
 - Initial version.

ADC

The current ADC driver version is 2.1.0.

- 2.1.0
 - Improvements
 - * Added the ADC_GetDefaultFIFOConfig() API to get default setting for FIFO configuration.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.3.
- 2.0.2
 - Bug Fixes
 - * Fixed the violations of MISRA C-2012 rules:
 - Rule 10.1 10.3 10.4 15.5 17.7.
- 2.0.1
 - Bug Fixes
 - * Fixed the missing right pair definition for extern C.
- 2.0.0
 - Initial version.

COMMON

The current COMMON driver version is 2.3.1.

- 2.3.1
 - Bug Fixes

- * Fixed MAKE_VERSION overflow on 16-bit platforms.
- 2.3.0
 - Improvements
 - * Split the driver to common part and CPU architecture related part.
- 2.2.10
 - Bug Fixes
 - * Fixed the ATOMIC macros build error in cpp files.
- 2.2.9
 - Bug Fixes
 - * Fixed MISRA C-2012 issue, 5.6, 5.8, 8.4, 8.5, 8.6, 10.1, 10.4, 17.7, 21.3.
 - * Fixed SDK_Malloc issue that not allocate memory with required size.
- 2.2.8
 - Improvements
 - * Included stddef.h header file for MDK tool chain.
 - New Features:
 - * Added atomic modification macros.
- 2.2.7
 - Other Change
 - * Added MECC status group definition.
- 2.2.6
 - Other Change
 - * Added more status group definition.
 - Bug Fixes
 - * Undef __VECTOR_TABLE to avoid duplicate definition in cmsis_clang.h
- 2.2.5
 - Bug Fixes
 - * Fixed MISRA C-2012 rule-15.5.
- 2.2.4
 - Bug Fixes
 - * Fixed MISRA C-2012 rule-10.4.
- 2.2.3
 - New Features
 - * Provided better accuracy of SDK_DelayAtLeastUs with DWT, use macro SDK_DELAY_USE_DWT to enable this feature.
 - * Modified the Cortex-M7 delay count divisor based on latest tests on RT series boards, this setting lets result be closer to actual delay time.
- 2.2.2
 - New Features
 - * Added include RTE_Components.h for CMSIS pack RTE.
- 2.2.1
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 Rule 3.1, 10.1, 10.3, 10.4, 11.6, 11.9.
- 2.2.0
 - New Features
 - * Moved SDK_DelayAtLeastUs function from clock driver to common driver.

- 2.1.4
 - New Features
 - * Added OTFAD into status group.
- 2.1.3
 - Bug Fixes
 - * MISRA C-2012 issue fixed.
 - Fixed the rule: rule-10.3.
- 2.1.2
 - Improvements
 - * Add SUPPRESS_FALL_THROUGH_WARNING() macro for the usage of suppressing fallthrough warning.
- 2.1.1
 - Bug Fixes
 - * Deleted and optimized repeated macro.
- 2.1.0
 - New Features
 - * Added IRQ operation for XCC toolchain.
 - * Added group IDs for newly supported drivers.
- 2.0.2
 - Bug Fixes
 - * MISRA C-2012 issue fixed.
 - Fixed the rule: rule-10.4.
- 2.0.1
 - Improvements
 - * Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
 - * Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTION" for specific SoCs which have no noncacheable sections, that helps avoid an unnecessary complex in link file and the startup file.
 - * Updated the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
 - Initial version.

FLASH

The current FLASH driver version is 2.1.2.

- 2.1.2
 - Improvements — The improved FLASH_EepromWrite function can write more data at once time.
- 2.1.1
 - Bug Fixes — MISRA C-2012 issue fixed: rule 14.4
- 2.1.0
 - New Features
 - * add feature macro before the declaration of the EEPROM_check_range.

- 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_SetupPwm-Mode() 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 GPIO driver version is 2.1.1.

- 2.1.1
 - Improvements:
 - * Enhanced FGPIO_PinInit to enable clock internally.

- 2.1.0
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 8.6.
 - * Updated parameter from base into port in port_init() API.
- 2.0.1
 - Bug Fixes
 - * Fixed violations of the MISRA C-2012 rules 3.1, 10.1, 10.3, 10.6, 10.7.
- 2.0.0
 - Initial version.

I2C

The current I2C driver version is 2.0.9.

- 2.0.9
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.4, 10.1, 10.4, 13.5, 20.8.
- 2.0.8
 - Bug Fixes
 - * Fixed the bug that DFEN bit of I2C Status register 2 could not be set in I2C_MasterInit.
 - * MISRA C-2012 issue fixed: rule 14.2, 15.7, and 16.4.
 - * Eliminated IAR Pa082 warnings from I2C_MasterTransferDMA and I2C_MasterTransferCallbackDMA by assigning volatile variables to local variables and using local variables instead.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 11.9, 14.4, 15.7, 17.7.
 - Improvements
 - * Improved timeout mechanism when waiting certain state in transfer API.
 - * Updated the I2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
 - * Moved the master manually acknowledge byte operation into static function I2C_MasterAckByte.
 - * Fixed control/status clean flow issue inside I2C_MasterReadBlocking to avoid potential issue that pending status is cleaned before it's proceeded.
- 2.0.7
 - Bug Fixes
 - * Fixed the issue for MISRA-2012 check.
 - Fixed rule 11.9 ,15.7 ,14.4 ,10.4 ,10.8 ,10.3, 10.1, 10.6, 13.5, 11.3, 13.2, 17.7, 5.7, 8.3, 8.5, 11.1, 16.1.
 - * Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
 - * Fixed variable redefine issue by moving i2cBases from fsl_i2c.h to fsl_i2c.c.
 - Improvements
 - * Added I2C_MASTER_FACK_CONTROL macro to enable FACK control for master transfer receive flow with IP supporting double buffer, then master could hold the SCL

by manually setting TX AK/NAK during data transfer.

- 2.0.6
 - Bug Fixes
 - * Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) did not support the situation of master transfer with subaddress and transfer data size being zero, which means no data followed by the subaddress.
- 2.0.5
 - Improvements
 - * Added I2C_WATL_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.4
 - Bug Fixes
 - * Added a proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Support write only or write+read with no start flag; does not support read only with no start flag.
- 2.0.3
 - Bug Fixes
 - * Removed enableHighDrive member in the master/slave configuration structure because the operation to HDRS bit is useless, the user need to use DSE bit in port register to configure the high drive capability.
 - * Added register reset operation in I2C_MasterInit and I2C_SlaveInit APIs. Fixed issue where I2C could not switch between master and slave mode.
 - * Improved slave IRQ handler to handle the corner case that stop flag and address match flag come synchronously.
- 2.0.2
 - Bug Fixes
 - * Fixed issue in master receive and slave transmit mode with no stop flag. The master could not succeed to start next transfer because the master could not send out re-start signal.
 - * Fixed the out-of-order issue of data transfer due to memory barrier.
 - * Added hold time configuration for slave. By leaving the SCL divider and MULT reset values when configured to slave mode, the setup and hold time of the slave is then reduced outside of spec for lower baudrates. This can cause intermittent arbitration loss on the master side.
 - New Features
 - * Added address nak event for master.
 - * Added general call event for slave.
- 2.0.1
 - New Features
 - * Added double buffer enable configuration for SoCs which have the DFEN bit in S2 register.
 - * Added flexible transmit/receive buffer size support in I2C_SlaveHandleIRQ.
 - * Added start flag clear, address match, and release bus operation in I2C_SlaveWrite/Read-Blocking API.
 - Bug Fixes
 - * Changed the kI2C_SlaveRepeatedStartEvent to kI2C_SlaveStartEvent.

- 2.0.0
 - Initial version.

IRQ

The current IRQ driver version is 2.0.2.

- 2.0.2
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 Rule 3.1, 8.4, 10.3 and 10.6.
- 2.0.1
 - New Features
 - * Added control macros to enable/disable the CLOCK code in current driver.
- 2.0.0
 - Initial version.

KBI

The current KBI driver version is 2.0.3.

- 2.0.3
 - Bug Fixes
 - * Fixed violation of the MISRA C-2012 rules 10.8.
- 2.0.2
 - Bug Fixes
 - * Fixed violations of the MISRA C-2012 rules 10.3.
- 2.0.1
 - Bug Fixes
 - * Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 17.7.
- 2.0.0
 - Initial version.

PIT

The current PIT driver version is 2.0.4.

- 2.0.4
 - Bug Fixes
 - * Fixed PIT_SetTimerPeriod implementation, the load value trigger should be PIT clock cycles minus 1.
- 2.0.3
 - Bug Fixes
 - * Clear all status bits for all channels to make sure the status of all TCTRL registers is clean.
- 2.0.2

- Bug Fixes
 - * Fixed MISRA-2012 issues.
 - Rule 10.1.
- 2.0.1
 - Bug Fixes
 - * Cleared timer enable bit for all channels in function PIT_Init() to make sure all channels stay in disable status before setting other configurations.
 - * Fixed MISRA-2012 rules.
 - Rule 14.4, rule 10.4.
- 2.0.0
 - Initial version.

PORT

The current PORT driver version is 2.2.0.

- 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, 10.4, 11.3, 11.8, 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 rule 17.7
- 2.0.0
 - Initial version.

RTC

The current RTC driver version is 2.0.4.

- 2.0.4
 - Improvements
 - * Changed the behavior of calling alarm callback when alarm seconds reach counter seconds, instead of previous behavior when counter seconds reach alarm seconds and counter seconds increments.
- 2.0.3
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 10.1, 10.3, 10.4 and 14.4.
- 2.0.2
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 10.3 and 11.9.
- 2.0.1
 - Bug Fixes
 - * Fixed the issue of Pa082 warning.
- 2.0.0
 - Initial version.

SPI

The current SPI driver version is 2.1.1.

- 2.1.1
 - Bug Fixes
 - * Fixed MISRA 10.3 violation.
- 2.1.0
 - Improvements
 - * Added timeout mechanism when waiting certain states in transfer driver.
 - Bug Fixes
 - * Fixed the bug that, when working as a slave, instance that does not have FIFO may miss some rx data.
 - * Fixed master RX data overflow issue by synchronizing transmit and receive process.
 - * Fixed issue that slave should not share the same non-blocking initialization API and IRQ handler with master to prevent dead lock issue.
 - * Fixed issue that callback should be invoked after all data is sent out to bus.
 - * Added code in SPI_SlaveTransferNonBlocking to empty rx buffer before initializing transfer.
- 2.0.5
 - Bug Fixes
 - * Eliminated Pa082 warnings from SPI_WriteNonBlocking and SPI_GetStatusFlags.
 - * Fixed MISRA issues.
 - Fixed issues 10.1, 10.3, 10.4, 10.7, 10.8, 11.9, 14.4, 17.7.

- 2.0.4
 - New Features
 - * Supported 3-wire mode for SPI driver. Added new API SPI_SetPinMode() to control the transfer direction of the single wire. For master instance, MOSI is selected as I/O pin. For slave instance, MISO is selected as I/O pin.
 - * Added dummy data setup API to allow users to configure the dummy data to be transferred.
- 2.0.3
 - Bug Fixes
 - * Fixed the potential interrupt race condition at high baudrate when calling API SPI_MasterTransferNonBlocking.
- 2.0.2
 - New Features
 - * Allowed users to set the transfer size for SPI_TransferNoBlocking non-integer times of watermark.
 - * Allowed users to define the dummy data. Users only need to define the macro SPI_DUMMYDATA in applications.
- 2.0.1
 - Bug Fixes
 - * Fixed SPI_Enable function parameter error.
 - * Set the s_dummy variable as static variable in fsl_spi_dma.c.
 - Improvements
 - * Optimized the code size while not using transactional API.
 - * Improved performance in polling method.
 - * Added #ifndef/#endif to allow users to change the default tx value at compile time.
- 2.0.0
 - Initial version.

TPM

The current TPM driver version is 2.2.0.

- 2.2.0
 - Improvements
 - * Added TPM_SetChannelPolarity to support select channel input/output polarity.
 - * Added TPM_EnableChannelExtTrigger to support enable external trigger input to be used by channel.
 - * Added TPM_CalculateCounterClkDiv to help calculates the counter clock prescaler.
 - * Added TPM_GetChannelValue to support get TPM channel value.
 - * Added new TPM configuration.
 - syncGlobalTimeBase
 - extTriggerPolarity
 - chnlPolarity
 - * Added new PWM signal configuration.

- secPauseLevel
- Bug Fixes
 - * Fixed TPM_SetupPwm can't configure 0% combined PWM issues.
- 2.1.1
 - Improvements
 - * Add feature macro for PWM pause level select feature.
- 2.1.0
 - Improvements
 - * Added TPM_EnableChannel and TPM_DisableChannel APIs.
 - * Added new PWM signal configuration.
 - pauseLevel
 - Support select output level when counter first enabled or paused.
 - enableComplementary
 - Support enable/disable generate complementary PWM signal.
 - deadTimeValue
 - Support deadtime insertion for each pair of channels in combined PWM mode.
 - Bug Fixes
 - * Fixed issues about channel MSnB:MSnA and ELSnB:ELSnA bit fields and CnV register change request acknowledgement. Writes to these bits are ignored when the interval between successive writes is less than the TPM clock period.
- 2.0.8
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.1, 10.4 ,10.7 and 14.4.
- 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_UpdateChnlEdgeLevelSelect ACK wait issue.
 - * Fixed the issue that TPM_SetupDualEdgeCapture could not set FILTER register.
 - * Fixed TPM_UpdateChnlEdgeLevelSelect ACK wait issue.
- 2.0.0
 - Initial version.

UART

The current UART driver version is 2.5.1.

- 2.5.1
 - Improvements
 - * Use separate data for TX and RX in uart_transfer_t.
 - Bug Fixes
 - * Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling UART_TransferReceiveNonBlocking, the received data count returned by UART_TransferGetReceiveCount is wrong.
- 2.5.0
 - New Features
 - * Added APIs UART_GetRxFifoCount/UART_GetTxFifoCount to get rx/tx FIFO data count.
 - * Added APIs UART_SetRxFifoWatermark/UART_SetTxFifoWatermark to set rx/tx FIFO water mark.
 - Bug Fixes
 - * Fixed bug of race condition during UART transfer using transactional APIs, by disabling and re-enabling the global interrupt before and after critical operations on interrupt enable registers.
 - * Fixed DMA/eDMA transfer blocking issue by enabling tx idle interrupt after DMA/eDMA transmission finishes.
- 2.4.0
 - New Features
 - * Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.0
 - Bug Fixes
 - * Fixed the bug that, when framing/parity/noise/overflow flag or idle line detect flag is set, receive FIFO should be flushed to avoid FIFO pointer being in unknown state, since FIFO has no valid data.
 - Improvements
 - * Modified UART_TransferHandleIRQ so that txState will be set to idle only when all

- data has been sent out to bus.
 - * Modified UART_TransferGetSendCount so that this API returns the real byte count that UART has sent out rather than the software buffer status.
 - * Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.0
 - New Features
 - * Added UART hardware FIFO enable/disable API.
 - Improvements
 - * Added check for kUART_TransmissionCompleteFlag in UART_TransferHandleIRQ, UART_SendEDMACallback and UART_TransferSendDMACallback to ensure all the data would be sent out to bus.
 - Bug Fixes
 - * Eliminated IAR Pa082 warnings from UART_TransferGetRxRingBufferLength, UART_GetEnabledInterrupts, UART_GetStatusFlags and UART_TransferHandleIRQ.
 - * Added code in UART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 14.4, 11.6, 17.7.
- 2.1.6
 - Bug Fixes
 - * Fixed the issue of register's being in repeatedly reading status while performing the IRQ routine.
- 2.1.5
 - Improvements
 - * Added hardware flow control function support.
 - * Added idle-line-detecting feature in UART_TransferNonBlocking function. If an idle line is detected, a callback will be triggered with status kStatus_UART_IdleLineDetected returned. This feature may be useful when the number of received bytes is less than the expected receive data size. Before triggering the callback, data in the FIFO is read out (if it has FIFO), and no interrupt will be disabled except for the case that the receive data size reaches 0.
 - * Enabled the RX FIFO watermark function. With the idle-line-detecting feature enabled, you can set the watermark value to whatever you want (should not be bigger than the RX FIFO size). Data is then received and a callback will be triggered when data receive ends.
- 2.1.4
 - Improvements
 - * Changed parameter type in UART_RTOS_Init() struct rtos_uart_config -> uart_rtos_config_t.
 - Bug Fixes
 - * Disabled UART receive interrupt instead of global interrupt when reading data from ring buffer. With ring buffer used, receive nonblocking will disable global interrupt to protect the ring buffer. This has a negative effect on other IPs using interrupt.
- 2.1.3
 - New Features

- * Added RX framing error and parity error status check when using interrupt transfer.
- 2.1.2
 - Bug Fixes
 - * Fixed baud rate fine adjust bug to make the computed baud rate more accurate.
- 2.1.1
 - Bug Fixes
 - * Removed needless check of event flags and assert in UART_RTOS_Receive.
 - * Always waited for RX event flag in UART_RTOS_Receive.
- 2.1.0
 - Improvements
 - * Added transactional API.
- 2.0.0
 - Initial version.

WDOG8

The current WDOG8 driver version is 2.0.1.

- 2.0.1
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 10.3, 10.4, 10.6, 10.7 and 11.9.
 - * Fixed the issue of the inseparable process interrupted by other interrupt source.
 - WDOG8_Refresh
- 2.0.0
 - 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.

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.

