

---

# **MCUXpresso SDK Release Notes Supporting Ipcxpresso55s16**

**Change Logs**

**NXP Semiconductors**



# Contents

## Component Change Log

<b>GENERICLIST</b> .....	<b>1</b>
<b>UART_ADAPTER</b> .....	<b>1</b>

## Driver Change Log

<b>COMMON</b> .....	<b>2</b>
<b>GPIO</b> .....	<b>4</b>
<b>LPTMR</b> .....	<b>5</b>
<b>LPUART_CMSIS</b> .....	<b>6</b>
<b>LPUART</b> .....	<b>6</b>
<b>LPUART_DMA</b> .....	<b>10</b>
<b>LPUART_FREERTOS</b> .....	<b>10</b>
<b>PORT</b> .....	<b>10</b>

# 1 Component Change Log

## GENERICLIST

The current GenericList component version is 1.0.1.

- 1.0.1
  - Add prefixing fsl\_component\_XXX.
- 1.0.0
  - Initial version

## UART\_ADAPTER

The current UART\_Adapter component version is 1.0.1.

- 1.0.1
  - Add prefixing fsl\_adapter\_XXX.
- 1.0.0
  - Initial version

## 2 Driver Change Log

### COMMON

The current COMMON driver version is 2.4.0.

- 2.4.0
  - New Features
    - \* Added EnableIRQWithPriority, IRQ\_SetPriority, and IRQ\_ClearPendingIRQ for ARM.
    - \* Added MSDK\_EnableCpuCycleCounter, MSDK\_GetCpuCycleCount for ARM.
- 2.3.3
  - New Features
    - \* Added NETC into status group.
- 2.3.2
  - Improvements
    - \* Make driver aarch64 compatible
- 2.3.1
  - Bug Fixes
    - \* Fixed MAKE\_VERSION overflow on 16-bit platforms.
- 2.3.0
  - Improvements
    - \* Split the driver to common part and CPU architecture related part.
- 2.2.10
  - Bug Fixes
    - \* Fixed the ATOMIC macros build error in cpp files.
- 2.2.9
  - Bug Fixes
    - \* Fixed MISRA C-2012 issue, 5.6, 5.8, 8.4, 8.5, 8.6, 10.1, 10.4, 17.7, 21.3.
    - \* Fixed SDK\_Malloc issue that not allocate memory with required size.
- 2.2.8
  - Improvements
    - \* Included stddef.h header file for MDK tool chain.
  - New Features:
    - \* Added atomic modification macros.
- 2.2.7
  - Other Change
    - \* Added MECC status group definition.
- 2.2.6
  - Other Change
    - \* Added more status group definition.
  - Bug Fixes
    - \* Undef \_\_VECTOR\_TABLE to avoid duplicate definition in cmsis\_clang.h
- 2.2.5
  - Bug Fixes

- \* Fixed MISRA C-2012 rule-15.5.
- 2.2.4
  - Bug Fixes
    - \* Fixed MISRA C-2012 rule-10.4.
- 2.2.3
  - New Features
    - \* Provided better accuracy of SDK\_DelayAtLeastUs with DWT, use macro SDK\_DELAY\_USE\_DWT to enable this feature.
    - \* Modified the Cortex-M7 delay count divisor based on latest tests on RT series boards, this setting lets result be closer to actual delay time.
- 2.2.2
  - New Features
    - \* Added include RTE\_Components.h for CMSIS pack RTE.
- 2.2.1
  - Bug Fixes
    - \* Fixed violation of MISRA C-2012 Rule 3.1, 10.1, 10.3, 10.4, 11.6, 11.9.
- 2.2.0
  - New Features
    - \* Moved SDK\_DelayAtLeastUs function from clock driver to common driver.
- 2.1.4
  - New Features
    - \* Added OTFAD into status group.
- 2.1.3
  - Bug Fixes
    - \* MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.3.
- 2.1.2
  - Improvements
    - \* Add SUPPRESS\_FALL\_THROUGH\_WARNING() macro for the usage of suppressing fallthrough warning.
- 2.1.1
  - Bug Fixes
    - \* Deleted and optimized repeated macro.
- 2.1.0
  - New Features
    - \* Added IRQ operation for XCC toolchain.
    - \* Added group IDs for newly supported drivers.
- 2.0.2
  - Bug Fixes
    - \* MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.4.
- 2.0.1
  - Improvements
    - \* Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
    - \* Added new feature macro switch "FSL\_FEATURE\_HAS\_NO\_NONCACHEABLE\_S-

CTION" 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.

## GPIO

The current driver version is 2.7.3.

- 2.7.3
  - Improvements
    - \* Release peripheral from reset if necessary in init function.
- 2.7.2
  - New Features
    - \* Support devices without PORT module.
- 2.7.1
  - Bug Fixes
    - \* Fixed MISRA C-2012 rule 10.4 issues in GPIO\_GpioGetInterruptChannelFlags() function and GPIO\_GpioClearInterruptChannelFlags() function.
- 2.7.0
  - New Features
    - \* Added API to support Interrupt select (IRQS) bitfield.
- 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".

## 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\_CMSIS

Current LPUART\_CMSIS driver version is 2.5

- 2.5
  - Improvement
    - \* Changed DMA\_Type to void for different platform dma.
- 2.4
  - Bug Fixes
    - \* Fixed the MISRA-2012 violations.
      - Fixed rule 10.3.
- 2.3
  - Other Changes
    - \* Update the edma request source to support more than 0xFF request sources.
- 2.2
  - Bug Fixes
    - \* Update driver to fix warnings reported by IAR v9.
- 2.1
  - Bug Fixes
    - \* Fixed the MISRA-2012 violations.
      - Fixed rule 8.4, 8.6, 10.1, 10.3, 10.4, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.0
  - Initial version.

## LPUART

The current LPUART driver version is 2.7.4.

- 2.7.4



- Improvements
  - \* Release peripheral from reset if necessary in init function.
- 2.7.3
  - Bug Fixes
    - \* Fixed violations of the MISRA C-2012 rules 15.7.
- 2.7.2
  - Bug Fix
    - \* Fixed the bug that the OSR calculation error when lpuart init and lpuart set baud rate.
- 2.7.1
  - Improvements
    - \* Added support for LPUART\_BASE\_PTRS\_NS in security mode in file fsl\_lpuart.c.
- 2.7.0
  - Improvements
    - \* Split some functions, fixed CCM problem in file fsl\_lpuart.c.
- 2.6.0
  - Bug Fixes
    - \* Fixed bug that when there are multiple lpuart instance, unable to support different ISR.
- 2.5.3
  - Bug Fixes
    - \* Fixed comments by replacing unused status flags kLPUART\_NoiseErrorInRxDataRegFlag and kLPUART\_ParityErrorInRxDataRegFlag with kLPUART\_NoiseErrorFlag and kLPUART\_ParityErrorFlag.
- 2.5.2
  - Bug Fixes
    - \* Fixed bug that when setting watermark for TX or RX FIFO, the value may exceed the maximum limit.
  - Improvements
    - \* Added check in LPUART\_TransferDMAHandleIRQ and LPUART\_TransferEdmaHandleIRQ to ensure if user enables any interrupts other than transfer complete interrupt, the dma transfer is not terminated by mistake.
- 2.5.1
  - Improvements
    - \* Use separate data for TX and RX in lpuart\_transfer\_t.
  - Bug Fixes
    - \* Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling LPUART\_TransferReceiveNonBlocking, the received data count returned by LPUART\_TransferGetReceiveCount is wrong.
- 2.5.0
  - Bug Fixes
    - \* Added missing interrupt enable masks kLPUART\_Match1InterruptEnable and kLPUART\_Match2InterruptEnable.
    - \* Fixed bug in LPUART\_EnableInterrupts, LPUART\_DisableInterrupts and LPUART\_GetEnabledInterrupts that the BAUD[LBKDIE] bit field should be soc specific.
    - \* Fixed bug in LPUART\_TransferHandleIRQ that idle line interrupt should be disabled when rx data size is zero.

- \* Deleted unused status flags kLPUART\_NoiseErrorInRxDataRegFlag and kLPUART\_ParityErrorInRxDataRegFlag, since firstly their function are the same as kLPUART\_NoiseErrorFlag and kLPUART\_ParityErrorFlag, secondly to obtain them one data word must be read out thus interfering with the receiving process.
- \* Fixed bug in LPUART\_GetStatusFlags that the STAT[LBKDIF], STAT[MA1F] and STAT[MA2F] should be soc specific.
- \* Fixed bug in LPUART\_ClearStatusFlags that tx/rx FIFO is reset by mistake when clearing flags.
- \* Fixed bug in LPUART\_TransferHandleIRQ that while clearing idle line flag the other bits should be masked in case other status bits be cleared by accident.
- \* Fixed bug of race condition during LPUART transfer using transactional APIs, by disabling and re-enabling the global interrupt before and after critical operations on interrupt enable register.
- \* Fixed DMA/eDMA transfer blocking issue by enabling tx idle interrupt after DMA/eDMA transmission finishes.
- New Features
  - \* Added APIs LPUART\_GetRxFifoCount/LPUART\_GetTxFifoCount to get rx/tx FIFO data count.
  - \* Added APIs LPUART\_SetRxFifoWatermark/LPUART\_SetTxFifoWatermark to set rx/tx FIFO water mark.
- 2.4.1
  - Bug Fixes
    - \* Fixed MISRA advisory 17.7 issues.
- 2.4.0
  - New Features
    - \* Added APIs to configure 9-bit data mode, set slave address and send address.
- 2.3.1
  - Bug Fixes
    - \* Fixed MISRA advisory 15.5 issues.
- 2.3.0
  - Improvements
    - \* Modified LPUART\_TransferHandleIRQ so that txState will be set to idle only when all data has been sent out to bus.
    - \* Modified LPUART\_TransferGetSendCount so that this API returns the real byte count that LPUART has sent out rather than the software buffer status.
    - \* Added timeout mechanism when waiting for certain states in transfer driver.
- 2.2.8
  - Bug Fixes
    - \* Fixed issue for MISRA-2012 check.
      - Fixed rule-10.3, rule-14.4, rule-15.5.
    - \* Eliminated Pa082 warnings by assigning volatile variables to local variables and using local variables instead.
    - \* Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 10.8, 14.4, 11.6, 17.7.
  - Improvements

- \* Added check for kLPUART\_TransmissionCompleteFlag in LPUART\_WriteBlocking, LPUART\_TransferHandleIRQ, LPUART\_TransferSendDMACallback and LPUART\_SendEDMACallback to ensure all the data would be sent out to bus.
- \* Rounded up the calculated sbr value in LPUART\_SetBaudRate and LPUART\_Init to achieve more 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 have a negative effect on other IPs that are using the interrupt.
- 2.2.2
  - Improvements
    - \* Added software reset feature support.
    - \* Added software reset API in LPUART\_Init.

- 2.2.1
  - Improvements
    - \* Added separate RX/TX IRQ number support.
- 2.2.0
  - Improvements
    - \* Added support of 7 data bits and MSB.
- 2.1.1
  - Improvements
    - \* Removed unnecessary check of event flags and assert in LPUART\_RTOS\_Receive.
    - \* Added code to always wait for RX event flag in LPUART\_RTOS\_Receive.
- 2.1.0
  - Improvements
    - \* Update transactional APIs.

## LPUART\_DMA

The current LPUART\_DMA driver version is 2.4.0.

- 2.4.0
  - Refer LPUART driver change log 2.1.0 to 2.4.0

## LPUART\_FREERTOS

The current LPUART\_FREERTOS driver version is 2.4.0.

- 2.4.0
  - Refer LPUART driver change log 2.1.0 to 2.4.0

## PORT

The current PORT driver version is 2.4.1.

- 2.4.1
  - Bug Fixes
    - \* Fixed the violations of MISRA C-2012 rules: 10.1, 10.8 and 14.4.
- 2.4.0
  - New Features
    - \* Updated port\_pin\_config\_t to support input buffer and input invert.
- 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.

**How to Reach Us:**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://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](http://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.

