

# **MCUXpresso SDK Release Notes**

## **Supporting MIMXRT1015-EVK**



# Contents

**Chapter 1 Overview..... 3**

**Chapter 2 MCUXpresso SDK..... 4**

**Chapter 3 Development tools..... 5**

**Chapter 4 Supported development systems.....6**

**Chapter 5 Release contents.....7**

**Chapter 6 MCUXpresso SDK release package.....8**

**Chapter 7 MISRA compliance..... 11**

**Chapter 8 Known issues..... 14**

# Chapter 1

## Overview

The MCUXpresso Software Development Kit (SDK) is a collection of software enablement for Microcontrollers that includes peripheral drivers, high-level stacks including USB and other middleware packages, such as multicore support and FatFs, and integrated RTOS support for FreeRTOS™ OS. In addition to the base enablement, the MCUXpresso SDK is augmented with demo applications and driver example projects, and API documentation to help the customers quickly leverage the support of the MCUXpresso SDK.

For more details about MCUXpresso SDK, see the MCUXpresso SDK homepage [MCUXpresso-SDK: Software Development Kit](#).

---

### NOTE

See the attached Change Logs section at the end of this document to reference the device-specific driver logs, middleware logs, and RTOS log.

---

## Chapter 2

# MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.5.0, includes support for both LPC and i.MX System-on-Chips (SoC). The same drivers, APIs, and middleware are still available with support for Kinetis, LPC, and i.MX silicon. The MCUXpresso SDK adds support for the MCUXpresso IDE, an Eclipse-based toolchain that works with all MCUXpresso SDKs. Easily import your SDK into the new toolchain to have access to all of the available components, examples, and demos for your target silicon. In addition to the MCUXpresso IDE, support for the MCUXpresso Config Tools allows for easy cloning of existing SDK examples and demos, allowing users to easily leverage the existing software examples provided by the SDK for their own projects.

### NOTE

In order to maintain compatibility with legacy Freescale code, the filenames and source code in MCUXpresso SDK containing the legacy Freescale prefix 'FSL' has been left as is. The 'FSL' prefix has been redefined as the NXP Foundation Software Library.

# Chapter 3

## Development tools

The MCUXpresso SDK was compiled and tested with these development tools:

- IAR Embedded Workbench for Arm version 8.32.3
- MDK-Arm Microcontroller Development Kit (Keil)<sup>®</sup> 5.26
- Makefiles support with GCC revision 7-2018-q2-update from Arm Embedded
- MCUXpresso IDE v10.3.1

## Chapter 4

# Supported development systems

This release supports boards and devices listed in this table. Boards and devices in boldface were tested in this release:

**Table 1. Supported MCU devices and development boards**

<b>Development boards</b>	<b>MCU devices</b>
<b>MIMXRT1015-EVK</b>	MIMXRT1015CAF4A, <b>MIMXRT1015DAF5A</b>

# Chapter 5

## Release contents

This table provides an overview of the MCUXpresso SDK release package contents and locations.

**Table 2. Release contents**

<b>Deliverable</b>	<b>Location</b>
Boards	<install_dir>/boards
Qualcomm WiFi	<install_dir>/middleware/wifi_qca
Demo applications	<install_dir>/boards/<board_name>/demo_apps
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
CMSIS driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Qualcomm WiFi stack examples	<install_dir>/boards/<board_name>/wifi_qca_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
Middleware	<install_dir>/middleware
USB stack	<install_dir>/middleware/usb
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Cortex Microcontroller Software Interface Standard (CMSIS) ARM Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

# Chapter 6

## MCUXpresso SDK release package

The MCUXpresso SDK release package contents are aligned with the silicon subfamily it supports. This includes the boards, CMSIS, devices, documentation, middleware, and RTOS support.

### 6.1 Device support

The device folder contains all available software enablement for the specific System-on-Chip (SoC) subfamily. This folder includes clock-specific implementation, device register header file, device register feature header file, CMSIS derived device SVD, and the system configuration source files. Included with the standard SoC support are folders containing peripheral drivers, toolchain support, and a simple debug console.

The device-specific header files provide a direct access to the MCU peripheral registers. The device header file provides an overall SoC memory mapped register definition. In addition to the overall device memory mapped header file, the MCUXpresso SDK also includes the feature header file for each peripheral instantiated on the SoC.

The toolchain folder contains the startup code and linker files for each supported toolchain. The startup code is a CMSIScompliant startup that efficiently transfers the code execution to the main() function.

#### 6.1.1 Board support

The boards folder provides the board-specific demo applications, driver examples, RTOS, and middleware examples.

#### 6.1.2 Demo applications and other examples

The demo applications demonstrate the usage of the peripheral drivers to achieve a system level solution. Each demo application contains a readme file that describes the operation of the demo and required setup steps.

The driver examples demonstrate the capabilities of the peripheral drivers. Each example implements a common use case to help demonstrate the driver functionality.

### 6.2 Middleware

#### 6.2.1 USB stack

See the *MCUXpresso SDK USB Stack User's Guide* (document MCUXSDKUSBSUG) for more information.

##### 6.2.1.1 Peripheral devices tested with USB Host stack

This table provides a list of USB devices tested with the USB Host stack.

**Table 3. Peripheral devices**

Device type	Device
-------------	--------

*Table continues on the next page...*



**Table 3. Peripheral devices (continued)**

USB HUB	BELKIN F5U233 BELKIN F5U304 BELKIN F5U307 BELKIN F4U040 UNITEK Y-2151 Z-TEK ZK032A HYUNDAI HY-HB608
USB flash drive	ADATA C008 32 GB ADATA S102 8 G ADATA S102 16 G Verbatim STORE N GO USB Device 8 G Kingston DataTraveler DT101 G2 SanDisk Cruzer Blade 8 GB Unisplendour 1 G Imation 2 GB V-mux 2 GB Sanmina-SCI 128 M Corporate Express 1 G TOSHIBA THUHYBS-008G 8 G Transcend JF700 8 G Netac U903 16 G SSK SFD205 8 GB Rex 4 GB SAMSUNG USB3.0 16GB
USB card reader/adapter	SSK TF adapter Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card

*Table continues on the next page...*

**Table 3. Peripheral devices (continued)**

USB Mouse	DELL MS111-P DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
USB Keyboard	DELL SK8135 DELL SK8115

## 6.2.2 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

## 6.2.3 CMSIS

The MCUXpresso SDK is shipped with the standard CMSIS development pack, including the prebuilt libraries.

# Chapter 7

## MISRA compliance

All MCUXpresso SDK drivers and USB stack comply to MISRA 2012 rules with the following exceptions.

**Table 4. MISRA exceptions**

Exception Rules	Description
Directive 4.4	Sections of code should not be commented out.
Directive 4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous.
Directive 4.6	Typedef that indicate size and signedness should be used in place of the basic numerical type.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a transaction unit then the implementation of the object should be hidden.
Directive 4.9	A function should be used in preference to a function like macro where they are interchangeable.
Directive 4.10	Precautions shall be taken in order to prevent the contents of a header file being included more than once.
Directive 4.11	The validity of values passed to library functions shall be checked.
Rule 2.3	A project should not contain unused type declarations.
Rule 2.4	A project should not contain unused tag declarations.
Rule 2.5	A project should not contain unused macro declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 3.1	The character sequences <code>/*</code> and <code>//</code> shall not be used within a comment.
Rule 5.1	External identifiers shall be distinct.
Rule 5.3	An identifier declared in an inner scope shall not hide an identifier declared in an outer scope.
Rule 5.7	A tag name shall be a unique identifier.
Rule 5.9	Identifiers that define objects or functions with external linkage shall be unique.
Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 8.3	All declarations of an object or function shall use the same names and type qualifiers.
Rule 8.6	An identifier with external linkage shall have exactly one external definition.
Rule 8.7	Octal constants shall not be used.

*Table continues on the next page...*

**Table 4. MISRA exceptions (continued)**

Rule 8.9	A object should be defined at block scope if its identified only appears in a single function.
Rule 10.1	Operands shall not be of an inappropriate essential type.
Rule 10.3	The value of an expression shall not be assigned to an object with a narrower essential type of a different essential type category.
Rule 10.4	Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential type.
Rule 10.6	The value of a composite expression shall not be assigned to an object with wider essential type.
Rule 10.7	If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type.
Rule 10.8	The value of a composite expression shall not be cast to a different essential type category or a wider essential type.
Rule 11.1	Conversions shall not be performed between a pointer to a function and any other type.
Rule 11.3	A case shall not be performed between a pointer to object type and a pointer to a different object type.
Rule 11.4	A conversion should not be performed between a pointer to object and an integer type.
Rule 11.5	A conversion should not be performed from pointer to void into pointer to object.
Rule 11.6	A cast shall not be performed between pointer to void and an arithmetic type.
Rule 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.2	The right hand operator of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand.
Rule 13.3	A full expression containing an increment(++ ) or decrement(-- ) operator should have no other potential side effects other than that caused by the increment or decrement operator.
Rule 13.5	The right hand operand of a logical && or    operator shall not contain persistent side effects.
Rule 14.2	A for loop shall be well formed.

*Table continues on the next page...*

**Table 4. MISRA exceptions (continued)**

Rule 14.4	The controlling expressions of an statement and the controlling expression of an iteration-statement shall have essentially Boolean type.
Rule 15.5	A function should have a single point of exit at the end.
Rule 16.1	All switch statements shall be well-formed.
Rule 17.1	The feature of <stdarg.h> shall not be used.
Rule 18.4	The +, -, += and -= operators should not be applied to an expression of pointer type.
Rule 19.2	The union keyword should not be used.
Rule 20.1	#include directives should only be preceded by preprocessor directives or comments.
Rule 20.10	The # and ## preprocessor operators should not be used.
Rule 21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.

## Chapter 8

# Known issues

### 8.1 Maximum file path length in Windows 7<sup>®</sup> Operating System

Windows 7 operating system imposes a 260 character maximum length for file paths. When installing the MCUXpresso SDK, place it in a directory close to the root to prevent file paths from exceeding the maximum character length specified by the Windows operating system. The recommended location is the C:\nxp folder.

### 8.2 Build fail for flexspi\_nor targets of MDK project

The build will fail for flexspi\_nor targets of the MDK project which have been cloned from the SDK package by MCUXpresso Config Tool. This issue will be fixed in the next release of the configuration tool.

The current workaround is to add the following settings in "Options for Target" -> "Linker" -> "Misc controls".

```
--keep=*(.boot_hdr.ivt)
--keep=*(.boot_hdr.boot_data)
--keep=*(.boot_hdr.conf)
--predefine="-DXIP_BOOT_HEADER_ENABLE=1"
```

### 8.3 J-Link V6.44 Device Selection window pops up when debugging on IAR 8.32.3

This issue is caused when the *JLinkDeviceName* in *IAR\_INSTALL\_FOLDER\arm\config\devices\NXP\i.MXi.MXRT\MIMXRT1015xxx5A.i79* is not aligned with the one in *SEGGER\_INSTALL\_FOLDER\JLink\_V644\JLinkDevices.xml*. To workaround this issue, select MIMXRT1015DAF5A in the J-Link V6.44 Target device settings.

---

# **MCUXpresso SDK Release Notes Supporting MIMXRT1015-EVK**

**Change Logs**

# Contents

<b>Driver Change Log</b>	<b>1</b>
<b>CLOCK</b> . . . . .	<b>1</b>
<b>IOMUXC</b> . . . . .	<b>1</b>
<b>LPI2C</b> . . . . .	<b>1</b>
<b>LPSPI_CMSIS</b> . . . . .	<b>2</b>
<b>LPUART</b> . . . . .	<b>2</b>
<b>ADC</b> . . . . .	<b>2</b>
<b>ADC_ETC</b> . . . . .	<b>2</b>
<b>AIPSTZ</b> . . . . .	<b>3</b>
<b>AOI</b> . . . . .	<b>3</b>
<b>BEE</b> . . . . .	<b>3</b>
<b>CACHE</b> . . . . .	<b>3</b>
<b>COMMON</b> . . . . .	<b>4</b>
<b>DCDC</b> . . . . .	<b>4</b>
<b>DMAMUX</b> . . . . .	<b>4</b>
<b>EDMA</b> . . . . .	<b>5</b>
<b>EWM</b> . . . . .	<b>6</b>
<b>FLEXIO</b> . . . . .	<b>6</b>
<b>FLEXIO_UART</b> . . . . .	<b>7</b>
<b>FLEXIO_I2C</b> . . . . .	<b>7</b>
<b>FLEXIO_SPI</b> . . . . .	<b>9</b>



# Contents

Title	Page Number
<b>FLEXIO_I2S</b> . . . . .	<b>9</b>
<b>FLEXIO_MCU_LCD</b> . . . . .	<b>10</b>
<b>FLEXIO_CAMERA</b> . . . . .	<b>10</b>
<b>FLEXRAM</b> . . . . .	<b>11</b>
<b>FLEXSPI</b> . . . . .	<b>11</b>
<b>GPC</b> . . . . .	<b>12</b>
<b>GPT</b> . . . . .	<b>12</b>
<b>GPIO</b> . . . . .	<b>12</b>
<b>KPP</b> . . . . .	<b>13</b>
<b>LPI2C</b> . . . . .	<b>13</b>
<b>LPSPI</b> . . . . .	<b>14</b>
<b>LPUART</b> . . . . .	<b>15</b>
<b>PIT</b> . . . . .	<b>16</b>
<b>PMU</b> . . . . .	<b>16</b>
<b>PWM</b> . . . . .	<b>16</b>
<b>QTMR</b> . . . . .	<b>16</b>
<b>RTWDOG</b> . . . . .	<b>16</b>
<b>SAI</b> . . . . .	<b>17</b>
<b>SPDIF</b> . . . . .	<b>19</b>
<b>SRC</b> . . . . .	<b>19</b>
<b>TEMPMON</b> . . . . .	<b>19</b>
<b>WDOG</b> . . . . .	<b>19</b>
<b>XBARA</b> . . . . .	<b>20</b>
<b>XBARB</b> . . . . .	<b>20</b>

# Contents

Title	Page Number
<b>Middleware Change Log</b>	<b>21</b>
FatFs for MCUXpresso SDK . . . . .	21
USB stack for MCUXpresso SDK . . . . .	21
QCA WiFi . . . . .	24
<b>RTOS Change Log</b>	<b>26</b>
FreeRTOS for MCUXpresso SDK . . . . .	26
FreeRTOS for MCUXpresso SDK . . . . .	26

# 1 Driver Change Log

## CLOCK

The current CLOCK driver version is 2.1.6.

- 2.1.6
  - Bug Fix:
    - \* Fix build issue with GCC compiler when include header from C++ file.
- 2.1.5
  - Bug Fix:
    - \* Add initialization of the fractional mode and spread spectrum mode in `CLOCK_InitSysPll()`.
- 2.1.4
  - Optimization:
    - \* Add `PerClk` in `clock_name_t` and `CLOCK_GetFreq`.
    - \* Add APIs to get the frequency of AHB clock and SEMC, IPG clock and PER clock.
- 2.1.3
  - Use double instead of `uint64_t` to achieve better performance with double precision FPU.
- 2.1.2
  - some minor fixes.
- 2.0.0
  - initial version.

## IOMUXC

The current IOMUXC driver version is 2.0.0.

- 2.0.0
  - initial version.
- 2.0.1
  - Delete enum value `kIOMUXC_GPR_USBExposureMode` in the `_iomuxc_gpr_mode`.

## LPI2C

Current LPI2C CMSIS driver version is 2.0

- 2.0
  - Initial version.

## **LPSPI\_CMSIS**

Current lpspi\_cmsis driver version is 2.1

- 2.1
  - Bug Fix:
    - \* Fix 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 feature:
    - \* Allow user to set up the default Transmit value by using ARM\_SPI\_SET\_DEFAULT\_TX\_VALUE. Please note that set default value is not supported in slave interrupts, because the pin will stay tristated if TX buffer is NULL.
    - \* Enable slave select mode in the new driver, but this has no effect when users set any of them because the driver can only support the hardware control function.
    - \* Enable 3-Wire mode, users can use ARM\_SPI\_MODE\_MASTER\_SIMPLEX/ARM\_SPI\_MODE\_SLAVE\_SIMPLEX to enable this feature. For ARM\_SPI\_MODE\_MASTER\_SIMPLEX mode, select SOUT pin 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.

## **LPUART**

Current LPUART CMSIS driver version is 2.0

- 2.0
  - Initial version.

## **ADC**

The current ADC driver version is 2.0.2.

- 2.0.2
  - Use conversion control feature macro instead of that in io map.
- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

## **ADC\_ETC**

The current ADC\_ETC driver version is 2.0.1.

- 2.0.1

- Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

## **AIPSTZ**

The current AIPSTZ driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **AOI**

The current AOI driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **BEE**

The current BEE driver version is 2.0.1.

- 2.0.1
  - Bug fix:
    - \* Fixed bug in key user key loading sequence. BEE must be enabled during loading of user key.
    - \* Fixed typos in comments.
  - Feature:
    - \* Setting of AES nonce was moved from BEE\_SetRegionKey() into separate BEE\_SetRegionNonce() function.
    - \* Changed handling of region settings. Both regions are configured simultaneously by BEE\_SetConfig() function. Configuration of FAC start and end address using IOMUXC\_GPRs was moved to application.
    - \* Added configuration setting for endian swap, access permission and region security level.
    - \* Default value for region address offset was changed to 0.
- 2.0.0
  - Initial version.

## **CACHE**

The current CACHE driver version is 2.0.1.

- 2.0.1

- Fixed cache size issue in L2CACHE\_GetDefaultConfig API.
- 2.0.0
  - Initial version.

## COMMON

The current COMMON driver version is 2.0.2.

- 2.0.2
  - MISRA C-2012 issue fixed.
    - \* Fixed rule contain: rule-10.4.
- 2.0.1
  - Remove the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
  - Add new feature macro switch "FSL\_FEATURE\_HAS\_NO\_NONCACHEABLE\_SECTION" for specific SoC which has no noncacheable sections, this will help avoid unnecessary complex in link file and startup file.
  - Update the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
  - Initial version.

## DCDC

The current DCDC driver version is 2.0.0.

- 2.0.0
  - Initial version.

## DMAMUX

The current DMAMUX driver version is 2.0.3.

- 2.0.3
  - Fix the issue for MISRA-2012 check.
    - \* Fixed rule 10.4,rule 10.3.
- 2.0.2
  - New feature:
    - \* Added an always-on enable feature to a DMA channel for ULP1 DMAMUX support.
- 2.0.1
  - Bug fix:
    - \* Fixed build warning while setting the DMA request source in DMAMUX\_SetSource-Change issue by changing the type of the parameter source from uint8\_t to uint32\_t.
- 2.0.0
  - Initial version.

## EDMA

The current eDMA driver version is 2.1.7.

- 2.1.7
  - Bug fix:
    - \* Fixed wrong transfer size setting
      - Add 8 bytes transfer configuration and feature for RT series
      - Add feature to support 16 bytes transfer for Kinetis
  - Bug fix:
    - \* Fixed the issue that EDMA\_HandleIRQ will go to incorrect branch When TCD is not used and callback function is not registered.
- 2.1.6
  - Bug fix:
    - \* Fixed KW3X MISRA Issue.
      - Rule 14.4, 10.8, 10.4, 10.7, 10.110.3, 13.5, 13.2.
  - Improvements:
    - \* Clear IRQ handler that not available for specific platform with macro FSL\_FEATURE\_EDMA\_MODULE\_CHANNEL\_IRQ\_ENTRY\_SHARED\_OFFSET.
- 2.1.5
  - Improvements:
    - \* Improve EDMA IRQ handler to support half interrupt feature.
- 2.1.4
  - Bug fix:
    - \* Clear enabled request, status during EDMA\_Init for the case that EDMA is halted before reinitialization.
- 2.1.3
  - Bug fix:
    - \* Add clear DONE bit in IRQ handler to avoid overwrite TCD issue.
    - \* Optimize above solution for the case that transfer request occurs in callback.
- 2.1.2
  - Improvements:
    - \* Added interface to get next TCD address.
    - \* Added interface to get the unused TCD number.
- 2.1.1
  - Improvements:
    - \* Added documentation for eDMA data flow when scatter/gather is implemented for the EDMA\_HandleIRQ API.
    - \* Updated and corrected some related comments in the EDMA\_HandleIRQ API and edma\_handle\_t struct.
- 2.1.0
  - Improvements:
    - \* Changed the EDMA\_GetRemainingBytes API into EDMA\_GetRemainingMajorLoopCount due to eDMA IP limitation (see API comments/note for further details).
- 2.0.5
  - Improvements:

- \* Added pubweak DriverIRQHandler for K32H844P (16 channels shared).
- 2.0.4
  - Improvements:
    - \* Added support for SoCs with multiple eDMA instances.
    - \* Added pubweak DriverIRQHandler for KL28T DMA1 and MCIMX7U5\_M4.
- 2.0.3
  - Bug fix:
    - \* Fixed the incorrect pubweak IRQHandler name issue, which causes re-definition build errors when client sets his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler.
- 2.0.2
  - Bug fix:
    - \* Fixed incorrect minorLoopBytes type definition in \_edma\_transfer\_config struct, and defined minorLoopBytes as uint32\_t instead of uint16\_t.
- 2.0.1
  - Bug fix:
    - \* Fixed the eDMA callback issue (which did not check valid status) in EDMA\_HandleIRQ API.
- 2.0.0
  - Initial version.

## **EWM**

The current EWM driver version is 2.0.1.

- 2.0.1
  - Fixed EWM\_Deinit hardfault issue.
- 2.0.0
  - Initial version.

## **FLEXIO**

The current FLEXIO driver version is 2.0.2.

- 2.0.2:
  - Improvements:
    - \* Split FlexIO component which combines all flexio/flexio\_uart/flexio\_i2c/flexio\_i2s drivers into several components. FlexIO component, flexio\_uart component, flexio\_i2c\_master component, and flexio\_i2s component.
- 2.0.1
  - Bug fix:
    - \* Fix the Dozen mode configuration error in FLEXIO\_Init API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.



## FLEXIO\_UART

The current FLEXIO\_UART driver version is 2.1.5.

- 2.1.5
  - Trigger the user callback when all data are in ring buffer are received when calling FLEXIO\_UART\_TransferReceiveNonBlocking.
- 2.1.4
  - Unify component full name to FLEXIO UART(DMA/EDMA) Driver
- 2.1.3
  - Bug fixes: The following modifications support FlexIO using multiple instances.
    - \* Removed FLEXIO\_Reset API in module Init APIs.
    - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.
    - \* Updated module Enable APIs to only support enable operation.
- 2.1.2
  - Bug fixes:
    - \* Fixed the transfer count calculation issue in FLEXIO\_UART\_TransferGetReceiveCount, FLEXIO\_UART\_TransferGetSendCount, FLEXIO\_UART\_TransferGetReceiveCountDMA, FLEXIO\_UART\_TransferGetSendCountDMA, FLEXIO\_UART\_TransferGetReceiveCountEDMA and FLEXIO\_UART\_TransferGetSendCountEDMA
    - \* Fixed the Dozen mode configuration error in FLEXIO\_UART\_Init API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.
    - \* Reported error when set baudrate too low and FLEXIO cannot reach that baudrate.
    - \* Disabled FLEXIO\_UART receive interrupt instead of disable all NVIC when read data from ring buffer. Because ring buffer is used, receive nonblocking disables all NVIC interrupts to protect the ring buffer. This has negative effects on other IPS which are using interrupt.
- 2.1.1
  - Bug fixes:
    - \* Changed the API name FLEXIO\_UART\_StopRingBuffer to FLEXIO\_UART\_TransferStopRingBuffer to align with the definition in C file.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.
    - \* Added txSize/rxSize in handle structure to record the transfer size.
  - Bug fixes:
    - \* Added error handle to handle the data count is zero or data buffer is NULL situation.

## FLEXIO\_I2C

The current FLEXIO\_I2C driver version is 2.1.7.

- 2.1.7

- Bug fix:
    - \* Fixed the issue that FLEXIO\_I2C\_MasterTransferBlocking does not wait for STOP bit sent.
- 2.1.6
  - Bug fix:
    - \* Fixed the issue that I2C Master transfer APIs (blocking/non-blocking) does not support the situation that master transfer with subaddress and transfer data size zero, which means no data follows by the subaddress.
- 2.1.5
  - Unify component full name to FLEXIO I2C Driver
- 2.1.4
  - Bug fixes: The following modifications support FlexIO using multiple instances.
    - \* Removed FLEXIO\_Reset API in module Init APIs.
    - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.
    - \* Updated module Enable APIs to only support enable operation.
- 2.1.3
  - Changed the prototype of FLEXIO\_I2C\_MasterInit to return kStatus\_Success if initialization successfully and return kStatus\_InvalidArgument if "(srcClock\_Hz / masterConfig->baudRate\_Bps) / 2 - 1" exceeds 0xFFU.
- 2.1.2
  - Fixed the FLEXIO I2C issue where the master cannot receive data from I2C slave in high baudrate.
  - Fixed the FLEXIO I2C issue where the master cannot receive NAK when master sends non-existent addr.
  - Fixed the FLEXIO I2C issue where the master cannot get transfer count successfully.
  - Fixed the FLEXIO I2C issue where the master cannot receive data successfully when sending data first.
  - Fixed the Dozen mode configuration error in FLEXIO\_I2C\_MasterInit API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.
  - Fixed the FLEXIO\_I2C\_MasterTransferBlocking API calls FLEXIO\_I2C\_MasterTransferCreateHandle issue. This leads the s\_flexioHandle/s\_flexioIsr/s\_flexioType variable written. Then, if calling FLEXIO\_I2C\_MasterTransferBlocking API multiple times, the s\_flexioHandle/s\_flexioIsr/s\_flexioType variable cannot be written anymore due to it being out of range. This leads to the following: NonBlocking transfer APIs cannot work due to register IRQ failed.
- 2.1.1
  - Bug fixes:
    - \* Implemented the FLEXIO\_I2C\_MasterTransferBlocking API which defined in header file but has no implementation in the C file.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.
    - \* Added transferSize in handle structure to record the transfer size.

## FLEXIO\_SPI

The current FLEXIO\_SPI driver version is 2.1.3.

- 2.1.3
  - Unify component full name to FLEXIO SPI(DMA/EDMA) Driver
- 2.1.2
  - Bug fixes: The following modification support FlexIO using multiple instances.
    - \* Removed FLEXIO\_Reset API in module Init APIs.
    - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.
    - \* Updated module Enable APIs to only support enable operation.
- 2.1.1
  - Bug fixes:
    - \* Fixed bug where FLEXIO SPI transfer data is in 16 bit per frame mode with eDMA.
    - \* Fixed bug where FLEXIO SPI transfer data is in 16 bit per frame and direction is Lsbfirst mode with eDMA and interrupt.
    - \* Fixed the Dozen mode configuration error in FLEXIO\_SPI\_MasterInit/FLEXIO\_SPI\_SlaveInit API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.
  - Optimization:
    - \* Added #ifndef/#endif to allow user to change the default tx value at compile time.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.
    - \* Added transferSize in handle structure to record the transfer size.
  - Bug fixes:
    - \* Fixed the error register address return for 16-bit data write in FLEXIO\_SPI\_GetTxData-RegisterAddress.
    - \* Provided independent IRQHandler/transfer APIs for Master and slave to fix the baudrate limit issue.

## FLEXIO\_I2S

The current FLEXIO\_I2S driver version is 2.1.6.

- 2.1.6
  - Bug fix:
    - \* Add reset flexio before flexio i2s init to make sure flexio status is normal.
- 2.1.5
  - Bug fix:
    - \* Fix i2s driver use hard code for bitwidth setting.
- 2.1.4
  - Unify component full name to FLEXIO I2S(DMA/EDMA) Driver
- 2.1.3

- Bug fixes: The following modifications support FlexIO using multiple instances.
  - \* Removed FLEXIO\_Reset API in module Init APIs.
  - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.
  - \* Updated module Enable APIs to only support enable operation.
- 2.1.2
  - New features:
    - \* Added configure items for all pin polarity and data valid polarity.
    - \* Added default configure for pin polarity and data valid polarity.
- 2.1.1
  - Bug fixes:
    - \* Fixed FlexIO I2S RX data read error and eDMA address error.
    - \* Fix FlexIO I2S slave timer compare setting error.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.
    - \* Added transferSize in handle structure to record the transfer size.

## **FLEXIO\_MCU\_LCD**

The current FLEXIO\_MCU\_LCD driver version is 2.0.2.

- 2.0.2
  - Unify component full name to FLEXIO\_MCU\_LCD(EDMA) Driver
- 2.0.1
  - Bug fixes: The following modification to support FlexIO using multiple instances.
    - \* Removed FLEXIO\_Reset API in module Init APIs.
    - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.
    - \* Updated module Enable APIs to only support enable operation.
- 2.0.0
  - Initial version.

## **FLEXIO\_CAMERA**

The current FLEXIO\_CAMERA driver version is 2.1.2.

- 2.1.2
  - Unify component full name to FLEXIO CAMERA(EDMA) Driver
- 2.1.1
  - Bug fixes: The following modifications support FlexIO using multiple instances.
    - \* Removed FLEXIO\_Reset API in module Init APIs.
    - \* Updated module Deinit APIs to reset the shifter/timer config instead of disable module and disable clock.

- \* Updated module Enable APIs to only support enable operation.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.

## **FLEXRAM**

The current FLEXRAM driver version is 2.0.4.

- 2.0.4
  - Fix FlexRAM driver missing extern C around functions in header file.
  - Remove magic address feature from driver.
- 2.0.3
  - Fix TCM size configuration is wrong when TCM bank number is not a value power of 2.
- 2.0.2
  - Updated driver due to Reference Manual update.
- 2.0.1
  - Fixed MISRA issue.
- 2.0.0
  - Initial version.

## **FLEXSPI**

The current FLEXSPI driver version is 2.0.5.

- 2.0.5
  - Bug fixes:
    - \* Fix FLEXSPI\_UpdateLUT cannot do partial update issue .
- 2.0.4
  - Bug fixes:
    - \* Reset flash size for all ports to zero 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 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:
    - \* Supports DQS write mask enable/disable feature during set FLEXSPI configuration.
    - \* Provides new API FLEXSPI\_TransferUpdateSizeEDMA for user to update eDMA transfer size(SSIZE/DSIZE) per DMA transfer.
  - Bug fixes:

- \* Fixed FLEXSPI\_Init invalid operation to enable AHB bus Read Access to IP RX FIFO issue.
- \* Fixed FLEXSPI\_Init incorrect operation to configure IP TX FIFO watermark issue.
- 2.0.1
  - Bug fixes:
    - \* Fixed the flag clear issue and AHB read Command index configuration issue in FLEXSPI\_SetFlashConfig.
    - \* Updated FLEXSPI\_UpdateLUT function to update LUT table from any index instead of previous command index.
    - \* Added bus idle wait in FLEXSPI\_SetFlashConfig and FLEXSPI\_UpdateLUT to ensure bus is idle before any change to FlexSPI controller.
    - \* Updated interrupt API FLEXSPI\_TransferNonBlocking and interrupt handle flow FLEXSPI\_TransferHandleIRQ.
    - \* Updated eDMA API FLEXSPI\_TransferEDMA.
- 2.0.0
  - Initial version.

## GPC

The current GPC driver version is 2.1.0.

- 2.1.0
  - Updated driver for IMX6RT.
- 2.0.0
  - Initial version.

## GPT

The current GPT driver version is 2.0.0.

- 2.0.0
  - Initial version.

## GPIO

The current GPIO driver version is 2.0.1.

- 2.0.1:
  - API interface changes:
    - \* Refined naming of API while keeping all original APIs, marking them as deprecated. Original API 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.

## KPP

The current KPP driver version is 2.0.0.

- 2.0.0
  - Initial version.

## LPI2C

The current LPI2C driver version is 2.1.6.

- 2.1.6
  - Bug fix:
    - \* Fix driver MISRA build error and C++ build error in LPI2C\_MasterSend and LPI2C\_SlaveSend.
- 2.1.5
  - Bug fix:
    - \* Extended the Driver IRQ handler to support LPI2C4 and change to use ARRAY\_SIZE(k-Lpi2cBases) instead of FEATURE COUNT to decide the array size for handle pointer array.
- 2.1.4
  - Bug fix:
    - \* Fixed the LPI2C\_MasterTransferEDMA receive issue when LPI2C share same request source for TX/RX DMA request. Previously, the API uses scatter-gather method, which handles the command transfer first, then handles the linked TCD which is preset with the receive data transfer. The issue is that the TX DMA request and the RX DMA request are both enabled, so when the DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happens due to TX FIFO empty. The result is the RX DMA transfer starts, without waiting on the expected RX DMA request.
    - \* Fixed the issue by enabling IntMajor interrupt for the command TCD and checking if there is a linked TCD to disable the TX DMA request in LPI2C\_MasterEDMACallback API.
- 2.1.3
  - Improvement:
    - \* 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 fix:
    - \* In LPI2C\_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag is detected.
- 2.1.1
  - Bug fix:
    - \* Disabled auto stop feature in eDMA driver. Previously, the auto-stop feature was enabled at transfer when transferring with stop flag. If the previous transfer was without stop flag,

because the auto stop feature is enabled, then when starting a new transfer with stop flag, the stop flag sends before starting the new transfer, and the start flag cannot successfully send, so the transfer cannot start.

- \* Changed default slave configuration with address stall false.
- 2.1.0
  - API name change:
    - \* 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.

## **LPSPI**

The current LPSPI driver version is 2.0.3.

- 2.0.3
  - Bug Fix:
    - \* Remove the LPSPI\_Reset() from LPSPI\_MasterInit() and LPSPI\_SlaveInit(), because this API may glitch the slave select line, if needed, please call this function manually.
- 2.0.2
  - New feature:
    - \* Added dummy data setup 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 fix:
    - \* 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 2.2.7.

- 2.2.7
  - Fix the 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
  - Fix the repeatedly reading status register issue while dealing with the IRQ routine.
- 2.2.5
  - Do not set or clear the TIE/RIE bits when using LPUART\_EnableTxDMA() and LPUART\_EnableRxDMA().
- 2.2.4
  - Added hardware flow control function support.
  - Added idle line detected feature in LPUART\_TransferNonBlocking function. If an idle line was 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 receive data size. Before triggering the callback, data in the FIFO (if has FIFO) is read out, and all interrupts will not be disabled, except if the receive data size reaches 0.
  - Enabled the RX FIFO watermark function. With the idle line detected feature enabled, you can set the watermark value to whatever you want (should be less than the RX FIFO size). Data is received and a callback is triggered when data receive is end.
- 2.2.3
  - Changed parameter type in LPUART\_RTOS\_Init() struct rtos\_lpuart\_config -> lpuart\_rtos\_config\_t.
  - Bug fix:
    - \* Disabled LPUART receive interrupt instead of disabling all NVIC when read data from ring buffer. Because the ring buffer is used, receive nonblocking disables all NVIC interrupts to protect the ring buffer. This has a negative effect to other IPS which are using the interrupt.
- 2.2.2
  - Added software reset feature support.
  - Added software reset API to LPUART\_Init().
- 2.2.1
  - Added separate RX/TX IRQ number support.
- 2.2.0
  - Added 7 data bits and MSB support.
- 2.1.1
  - Removed needless check of event flags and assert in LPUART\_RTOS\_Receive.
  - Always wait for RX event flag in LPUART\_RTOS\_Receive.
- 2.1.0
  - Update transactional APIs.

## **PIT**

The current PIT driver version is 2.0.2.

- 2.0.2
  - Bug Fix:
    - \* Fix MISRA-2012 issues.
      - Rule 10.1.
- 2.0.1
  - Bug Fix:
    - \* Clear timer enable bit for all channels in function PIT\_Init() to make sure all channels stay in disable status before setting other configurations.
    - \* Fix MISRA-2012 rules.
      - Rule 14.4, rule 10.4.
- 2.0.0
  - Initial version.

## **PMU**

The current PMU driver version is 2.1.0.

- 2.1.0
  - Added feature macros for low power control APIs to support to conditional compile.
  - Renamed "PMU\_2P1EnablePullDown" to "PMU\_2P5EnablePullDown"
- 2.0.0
  - Initial version.

## **PWM**

The current PWM driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **QTMR**

The current QTMR driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **RTWDOG**

The current RTWDOG driver version is 2.1.0.

- 2.1.0
  - Add api to enable or disable the window mode.
  - Add api to convert a raw count value to millisecond.
- 2.0.1
  - Fix the bug in the RTWDOG\_Init, add the check of register's unlock status when configure the RTWDOG in RTWDOG\_init.
- 2.0.0
  - Initial version.

## SAI

The current SAI driver version is 2.2.0.

- 2.2.0
  - Improvement:
    - \* Add new APIs for parameters collection and user interfaces simplify: SAI\_Init  
SAI\_SetMasterClockConfig  
SAI\_TxSetBitClockRate SAI\_TxSetSerialDataConfig SAI\_TxSetFrameSyncConfig S-  
AI\_TxSetFifoConfig SAI\_TxSetBitclockConfig SAI\_TxSetConfig SAI\_TxSetTransfer-  
Config  
SAI\_RxSetBitClockRate SAI\_RxSetSerialDataConfig SAI\_RxSetFrameSyncConfig S-  
AI\_RxSetFifoConfig SAI\_RxSetBitclockConfig SAI\_RXSetConfig SAI\_RxSetTransfer-  
Config  
SAI\_GetClassicI2SConfig SAI\_GetLeftJustifiedConfig SAI\_GetRightJustifiedConfig S-  
AI\_GetTDMConfig
- 2.1.9
  - Improvement:
    - \* Improve SAI driver comment for clock polarity.
    - \* Add enum for SAI for sample inputs on different edge.
    - \* Change FSL\_FEATURE\_SAI\_CHANNEL\_COUNT to FSL\_FEATURE\_SAI\_CHANN-  
EL\_COUNTn(base) for the difference between the different SAI instance.
- 2.1.8
  - Improvement:
    - \* Add feature macro test for the sync mode2 and mode 3.
    - \* Add feature macro test for masterClockHz in sai\_transfer\_format\_t.
- 2.1.7
  - Improvement:
    - \* Add feature macro test for the mclkSource member in sai\_config\_t.
    - \* Change "FSL\_FEATURE\_SAI5\_SAI6\_SHARE\_IRQ" to "FSL\_FEATURE\_SAI\_SAI5-  
\_SAI6\_SHARE\_IRQ".
    - \* Add #ifndef #endif check for SAI\_XFER\_QUEUE\_SIZE to allow redefinition.
  - Bug fix:
    - \* Fix the build error caused by feature macro test for mclkSource.

-2.1.6

- Improvement:
  - Add feature macro test for mclkSourceClockHz check.
  - Add bit clock source name for general devices.
- Bug fix:
  - Fix incorrect channel numbers setting while call RX/TX set format together.

#### -2.1.5

- Bug fix:
  - Correct SAI3 driver IRQ handler name.
  - Add I2S4/5/6 IRQ handler.
  - Add base in handler structure to support different instances share one IRQ number.
- New feature:
  - Update SAI driver for MCR bit MICS.
  - Added 192KHZ/384KHZ in the sample rate enumeration.
  - Added multi FIFO interrupt/SDMA transfer support for TX/RX.
  - Added 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 feature:
  - Added API to enable/disable auto FIFO error recovery in platforms that support this feature.
  - Added API to set data packing feature in platform which support this feature.

#### 2.1.3

- New feature:
  - Added feature to make I2S frame sync length configurable according to bitWidth.

#### 2.1.2

- Bug fix:
  - 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

- Optimization:
  - Reduced code size while not using transactional API.

#### 2.1.0

- API name change:
  - SAI\_GetSendRemainingBytes -> SAI\_GetSentCount.
  - SAI\_GetReceiveRemainingBytes -> SAI\_GetReceivedCount.
  - All transactional API name add "Transfer" prefix.
  - All transactional API use base and handle as input parameter.
  - Unify the parameter names.
- Bug fix:
  - Fixed WLC bug while reading TCSR/RCSR registers.
  - Fixed MOE enable flow issue, move MOE enable after MICS settings in SAI\_TxInit/SAI\_Rx-

Init.

2.0.0

- Initial version.

## **SPDIF**

The current SPDIF driver version is 2.0.1.

- 2.0.1
  - Correct the feature macro name used to define s\_edmaPrivateHandle.
- 2.0.0
  - Initial version.

## **SRC**

The current SRC driver version is 2.0.1.

- 2.0.1
  - Improvement Update src driver for adding SRC\_SRSR\_JTAG\_SW\_RST enumeration.
- 2.0.0
  - Initial version.

## **TEMPMON**

The current TEMPMON driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **WDOG**

The current WDOG driver version is 2.1.0.

- 2.1.0
  - New Feature:
    - \* Add new API "WDOG\_TriggerSystemSoftwareReset()" to allow user reset the system by software.
    - \* Add new API "WDOG\_TriggerSoftwareSignal()" to allow user trigger a WDOG\_B signal by software.
    - \* Remove the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog\_config\_t structure.
    - \* Add new parameter "enableTimeOutAssert" to the wdog\_config\_t structure, with this parameter enabled, while the WDOG timeout occurred, a WDOG\_B signal will be asserted,

this signal can be routed to external pin of the chip. Please note that, WDOG\_B signal remains asserted until a power-on reset (POR) occurs.

- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

## **XBARA**

The current XBARA driver version is 2.0.4.

- 2.0.4
  - Improvement:
    - \* Optimize XBARA\_SetOutputSignalConfig.
- 2.0.3
  - Bug fixes:
    - \* Corrected configuration for function XBAR\_SetOutputSignalConfig.
- 2.0.2
  - Other changes:
    - \* Changed array clock name.
- 2.0.1
  - Bug fixes:
    - \* Fixed w1c bits for XBARA\_SetOutputSignalConfig function.
- 2.0.0
  - Initial version.

## **XBARB**

The current XBARB driver version is 2.0.1.

- 2.0.1
  - Bug fixes:
    - \* Corrected XBARB\_SetSignalsConnection function.
  - Other changes:
    - \* Changed array clock name.
- 2.0.0
  - Initial version.

## 2 Middleware Change Log

### FatFs for MCUXpresso SDK

Current version is FatFs R0.13b\_rev0.

- R0.13b\_rev0
  - Upgraded to version 0.13b
- R0.13a\_rev0
  - Upgraded to version 0.13a. Added patch ff\_13a\_p1.diff.
- R0.12c\_rev1
  - Add NAND disk support.
- R0.12c\_rev0
  - Upgraded to version 0.12c and applied patches ff\_12c\_p1.diff and ff\_12c\_p2.diff.
- R0.12b\_rev0
  - Upgraded to version 0.12b.
- R0.11a
  - Added glue functions for low-level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c.
  - Added RTOS wrappers to make FatFs thread safe. Modified syscall.c.
  - Renamed ffconf.h to ffconf\_template.h. Each application should contain its own ffconf.h.
  - Included ffconf.h into diskio.c to enable the selection of physical disk from ffconf.h by macro definition.
  - Conditional compilation of physical disk interfaces in diskio.c.

### USB stack for MCUXpresso SDK

The current version of USB stack is 2.1.0.

- 2.1.0
  - New features:
    - \* add host RNDIS support. example: lwip\_dhcp\_usb
    - \* enable USB 3.0 support on device stack.
    - \* Power Delivery feature: Add OM13790HOST support; Add auto policy feature; Print e-marked cable information;
- 2.0.1
  - Bug fix:
    - \* fixed some USB issues: fix msc cv test failed in msc examples.
    - \* Change the audio codec interfaces.
- 2.0.0
  - New features:
    - \* PTN5110N support.
  - Bug fix:
    - \* Added some comments, fixed some minor USB issues.
- 1.9.0

- New features:
    - \* Examples:
      - usb\_pd\_alt\_mode\_dp\_host
- 1.8.2
  - Updated license.
- 1.8.1
  - Bug fix:
    - \* Verified some hardware issues, support aruba\_flashless.
- 1.8.0
  - New features:
    - \* Examples:
      - usb\_device\_composite\_cdc\_vcom\_cdc\_vcom
      - usb\_device\_composite\_hid\_audio\_unified
      - usb\_pd\_sink\_battery
      - Changed usb\_pd\_battery to usb\_pd\_charger\_battery.
- Bug fix:
  - Code clean up, removed some irrelevant code.
- 1.7.0
  - New features:
    - \* USB PD stack support.
  - Examples
    - \* usb\_pd
    - \* usb\_pd\_battery
    - \* usb\_pd\_source\_charger
- 1.6.3
  - Bug fix: -IP3511\_HS driver control transfer sequence issue, enabled 3511 ip cv test.
- 1.6.2
  - New features:
    - \* Multi instance support.
- 1.6.1
  - New features:
    - Changed the struct variable address method for device\_video\_virtual\_camera and host\_phdc\_manager.
- 1.6.0
  - New features:
    - \* Supported Device Charger Detect feature on usb\_device\_hid\_mouse.
- 1.5.0
  - New features:
    - \* Supported controllers
      - OHCI (Full Speed, Host mode)
      - IP3516 (High Speed, Host mode)
      - IP3511 (High Speed, Device mode)
    - \* Examples:



- usb\_lpm\_device\_hid\_mouse
  - usb\_lpm\_device\_hid\_mouse\_lite
  - usb\_lpm\_host\_hid\_mouse
- 1.4.0
  - New features:
    - \* Examples:
      - usb\_device\_hid\_mouse/freertos\_static
      - usb\_suspend\_resume\_device\_hid\_mouse\_lite
- 1.3.0
  - New features:
    - \* Supported roles
      - OTG
    - \* Supported classes
      - CDC RNDIS
    - \* Examples
      - usb\_otg\_hid\_mouse
      - usb\_device\_cdc\_vnic
      - usb\_suspend\_resume\_device\_hid\_mouse
      - usb\_suspend\_resume\_host\_hid\_mouse
- 1.2.0
  - New features:
    - \* Supported controllers
      - LPC IP3511 (Full Speed, Device mode)
- 1.1.0
  - Bug fix:
    - \* Fixed some issues in USB certification.
    - \* Changed VID and Manufacturer string to NXP.
  - New features:
    - \* Supported classes
      - Pinter
    - \* Examples:
      - usb\_device\_composite\_cdc\_msc\_sdcard
      - usb\_device\_printer\_virtual\_plain\_text
      - usb\_host\_printer\_plain\_text
- 1.0.1
  - Bug fix:
    - \* Improved the efficiency of device audio speaker by changing the transfer mode from interrupt to DMA, thus providing the ability to eliminate the periodic noise.
- 1.0.0
  - New features:
    - \* Supported roles
      - Device
      - Host
    - \* Supported controllers:
      - KHCI (Full Speed)

- EHCI (High Speed)
- \* Supported classes:
  - AUDIO
  - CCID
  - CDC
  - HID
  - MSC
  - PHDC
  - VIDEO
- \* Examples:
  - usb\_device\_audio\_generator
  - usb\_device\_audio\_speaker
  - usb\_device\_ccid\_smart\_card
  - usb\_device\_cdc\_vcom
  - usb\_device\_cdc\_vnic
  - usb\_device\_composite\_cdc\_msc
  - usb\_device\_composite\_hid\_audio
  - usb\_device\_composite\_hid\_mouse\_hid\_keyboard
  - usb\_device\_hid\_generic
  - usb\_device\_hid\_mouse
  - usb\_device\_msc\_ramdisk
  - usb\_device\_msc\_sdcard
  - usb\_device\_phdc\_weighscale
  - usb\_device\_video\_flexio\_ov7670
  - usb\_device\_video\_virtual\_camera
  - usb\_host\_audio\_speaker
  - usb\_host\_cdc
  - usb\_host\_hid\_generic
  - usb\_host\_hid\_mouse
  - usb\_host\_hid\_mouse\_keyboard
  - usb\_host\_msd\_command
  - usb\_host\_msd\_fatfs
  - usb\_host\_phdc\_manager
  - usb\_keyboard2mouse
  - usb\_pin\_detect\_hid\_mouse

## **QCA WiFi**

The current version is 2.0.0.

- 2.0.0
  - Initial version.
    - \* Added QCA WiFi, ported from SDK 1.3, synchronized with latest MQX Qualcomm v3.-3.5.

- Known issues:
  - \* Low power mode may not work, require further investigation.
  - \* DHCP request requires some timeout to retrieve valid data.

### 3 RTOS Change Log

#### FreeRTOS for MCUXpresso SDK

The current version is Amazon-FreeRTOS 1.4.0. Original package is available at [github.com/aws/amazon-freertos](https://github.com/aws/amazon-freertos).

- 1.4.0
  - New features:
    - \* Support for CM33, CM33F architectures based on CM3, CM4F ports
    - \* 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

#### FreeRTOS for MCUXpresso SDK

The current version is FreeRTOS 9.0.0. Original package is available at [freertos.org](https://freertos.org).

- 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.

## 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 reserves the right to make changes without further notice to any products herein.

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, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro,  $\mu$ Vision, Versatile 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.

© NXP B.V. 2019.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: March 2019

Document identifier: MCUXSDKMIMXRT1015RN

