

# MCUXpresso SDK Release Notes Supporting MAPS-KS22

## 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 lwIP, integration with WolfSSL and mbed TLS cryptography libraries, 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 the latest version of this and other MCUXpresso SDK documents, 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.

## 2 MCUXpresso SDK

### Contents

1	Overview.....	1
2	MCUXpresso SDK.....	1
3	Development tools.....	2
4	Supported development systems.....	2
5	Release contents.....	2
6	MCUXpresso SDK release package.....	3
7	MISRA compliance.....	6
8	Known issues.....	8



## Development tools

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.3.0, 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, a new 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 FSL 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.

## 3 Development tools

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

- IAR Embedded Workbench for Arm version 8.22.2
- MDK-Arm Microcontroller Development Kit (Keil)<sup>®</sup> 5.24a
- Makefiles support with GCC revision 7-2017-q4-major from Arm Embedded
- MCUXpresso IDE v10.2.0

## 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**

Development boards	MCU devices
<b>MAPS-KS22</b>	<b>MKS22FN256VLL12</b> , MKS22FN256VLH12, MKS22FN128VLL12, MKS22FN128VLH12

## 5 Release contents

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

**Table 2. Release contents**

Deliverable	Location
Boards	<install_dir>/boards
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
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples

*Table continues on the next page...*

**Table 2. Release contents (continued)**

Multicore examples	<install_dir>/boards/<board_name>/multiprocessor_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
lwIP Documentation	<install_dir>/docs/lwip
Middleware	<install_dir>/middleware
lwIP stack	<install_dir>/middleware/lwip
DMA manager	<install_dir>/middleware/dma_manager
EMV stack	<install_dir>/middleware/emv
FatFS stack	<install_dir>/middleware/fatfs
mmCAU	<install_dir>/middleware/mmcau
Motor Control libraries	<install_dir>/middleware/motor_control
Multicore stack	<install_dir>/middleware/multicore
RTCESL libraries	<install_dir>/middleware/rtcesl
SDMMC card driver	<install_dir>/middleware/sdmmc
USB stack	<install_dir>/middleware/usb
WolfSSL stack	<install_dir>/middleware/wolfssl
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
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools

## 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 CMSIS-compliant 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.

The RTOS and middleware folders each contain examples demonstrating the use of the included source.

## 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 the USB Host stack

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

**Table 3. Peripheral devices**

Device type	Device
USB HUB	BELKIN F5U233
	BELKIN F5U304
	BELKIN F5U307
	BELKIN F4U040
	UNITEK Y-2151
	Z-TEK ZK032A
USB flash drive	HYUNDAI HY-HB608
	ADATA C008 32 GB
	ADATA S102 8 G
	ADATA S102 16 G
	Verbatim STORE N GO USB Device 8 G

*Table continues on the next page...*

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

	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/adaptor	SSK TF adapter Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card
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 TCP/IP stack

The lwIP TCP/IP stack is pre-integrated with MCUXpresso SDK and runs on top of the MCUXpresso SDK Ethernet driver with Ethernet-capable devices/boards. For details, see the *lwIP TCP/IP Stack and MCUXpresso SDK Integration User's Guide* (document MCUXSDKLWIPUG).

## 6.2.3 File system

The FatFs file system is integrated with MCUXpresso SDK and can be used to access either the SD card or the USB memory stick when the SD card driver or the USB Mass Storage Device class implementation is used.

## 6.2.4 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

## 6.2.5 CMSIS

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

# 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 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 /* and // shall not be used within a comment.
Rule 5.1	External identifiers shall distinct.
Rule 5.3	A 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.

*Table continues on the next page...*

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

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.
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 a 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 17.7	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.

## 8 Known issues

### 8.1 Maximum file path length in Windows® 7 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 USBFS controller issue

Because of the USBFS controller design issues, the USB host suspend/resume demos (usb\_suspend\_resume\_host\_hid\_mouse) of the full speed controller do not support the low speed device directly.

### 8.3 USB PID issue

Because the PID of all USB device examples is updated, uninstall the device drivers and then reinstall when the device (with new PID) is plugged in the first time.

**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).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, Freescale, the Freescale logo, Kinetis, and Tower are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex, Keil, and  $\mu$ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2018 NXP B.V.

Document Number MCUXSDKKS22RN  
Revision 0, 05/2018



---

# **MCUXpresso SDK Release Notes Supporting MAPS-KS22**

**Change Logs**

# Contents

<b>Driver Change Log</b>	<b>1</b>
<b>ADC16</b> . . . . .	<b>1</b>
<b>CMP</b> . . . . .	<b>1</b>
<b>CRC</b> . . . . .	<b>1</b>
<b>DAC</b> . . . . .	<b>1</b>
<b>DMAMUX</b> . . . . .	<b>1</b>
<b>DSPI</b> . . . . .	<b>2</b>
<b>EDMA</b> . . . . .	<b>3</b>
<b>EWM</b> . . . . .	<b>4</b>
<b>FLASH</b> . . . . .	<b>4</b>
<b>FLEXCAN</b> . . . . .	<b>5</b>
<b>FLEXIO</b> . . . . .	<b>5</b>
<b>FLEXIO_UART</b> . . . . .	<b>5</b>
<b>FLEXIO_I2C</b> . . . . .	<b>6</b>
<b>FLEXIO_SPI</b> . . . . .	<b>7</b>
<b>FLEXIO_I2S</b> . . . . .	<b>8</b>
<b>FLEXIO_MCU_LCD</b> . . . . .	<b>8</b>
<b>FLEXIO_CAMERA</b> . . . . .	<b>9</b>
<b>GPIO</b> . . . . .	<b>9</b>
<b>LLWU</b> . . . . .	<b>10</b>
<b>LPI2C</b> . . . . .	<b>10</b>

# Contents

	Page Number
<b>LPTMR</b> . . . . .	11
<b>LPUART</b> . . . . .	11
<b>PDB</b> . . . . .	12
<b>PIT</b> . . . . .	12
<b>PMC</b> . . . . .	12
<b>PORT</b> . . . . .	13
<b>RCM</b> . . . . .	13
<b>RTC</b> . . . . .	13
<b>SAI</b> . . . . .	13
<b>SIM</b> . . . . .	14
<b>SMARTCARD</b> . . . . .	14
<b>SMC</b> . . . . .	15
<b>TPM</b> . . . . .	15
<b>UART</b> . . . . .	16
<b>WDOG</b> . . . . .	16
<b>CLOCK</b> . . . . .	17
<b>Middleware Change Log</b>	18
<b>DMA_MANAGER</b> . . . . .	18
<b>EMVL1 for MCUXpresso SDK</b> . . . . .	18
<b>FatFs for MCUXpresso SDK</b> . . . . .	18
<b>SDMMC</b> . . . . .	19
<b>USB stack for MCUXpresso SDK</b> . . . . .	21
<b>RTOS Change Log</b>	25
<b>FreeRTOS for MCUXpresso SDK</b> . . . . .	25

## **1 Driver Change Log**

### **ADC16**

The current ADC16 driver version is 2.0.0.

- 2.0.0
  - Initial version

### **CMP**

The current CMP driver version is 2.0.0.

- 2.0.0
  - Initial version.

### **CRC**

The current CRC driver version is 2.0.1.

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

### **DAC**

The current DAC driver version is 2.0.1.

- 2.0.1
  - Bug fix:
    - \* Moved the default DAC\_Enable(..., true) from DAC\_Init() to the application code so users can enable the DAC's output.

2.0.0

- Initial version.

### **DMAMUX**

The current DMAMUX driver version is 2.0.2.

- 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.

## DSPI

The current dspi driver version is 2.2.0.

- 2.2.0
  - New features:
    - \* Added gasket feature for SPI EDMA driver, which reduces one channel used in the EDMA master transfer. With this feature support, only two channels are needed. For example, if the gasket feature is supported, we could use the DSPI\_MasterTransferCreateHandleEDMA function like below: DSPI\_MasterTransferCreateHandleEDMA(EXAMPLE\_DSPI\_MASTER\_BASEADDR, &g\_dspi\_edma\_m\_handle, DSPI\_MasterUserCallback, &userData, &dspiEdmaMasterRxRegToRxDataHandle, NULL, &dspiEdmaMasterIntermediaryToTxRegHandle);
    - \* Added dummy data setup API to allow users to configure the dummy data to be transferred.
    - \* Added new APIs for half-duplex transfer function. Users can send and receive data by one API in the polling/interrupt/EDMA way, and users can choose to either transmit first or receive first. Additionally, the PCS pin can be configured as assert status in transmission (between transmit and receive) by setting the isPcsAssertInTransfer to true.
- 2.1.4
  - Bug fix:
    - \* DSPI EDMA driver: The DSPI instance that has separated so the DMA request source can now transfer up to 32767 Bytes data in one DSPI\_MasterTransferEDMA() transfer.
- 2.1.3
  - Bug fix:
    - \* DSPI EDMA driver can no longer support the case that the transfer data size is odd, but the bitsPerFrame is greater than 8.
  - Optimization:
    - \* Added #ifndef/#endif to allow users to change the default TX value at compile time.
- 2.1.2
  - Bug fix:
    - \* DSPI\_MasterTransferBlocking function would hang in some corner cases (for example, some cases with bitsPerFrame is 4,6 and kDSPI\_MasterPcsContinuous transfer mode).
- 2.1.1
  - Bug fix:

- \* Set the EOQ (End Of Queue) bit to TRUE for the last transfer in transactional APIs.
- 2.1.0
  - New features:
    - \* Added Transfer prefix in transactional APIs.

## EDMA

The current eDMA driver version is 2.1.2.

- 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 wrong 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.

## FLASH

The current FLASH driver version is 2.3.1.

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

## FLEXCAN

The current FLEXCAN driver version is 2.2.0.

- 2.2.0
  - Improvements:
    - \* Added FSL\_FEATURE\_FLEXCAN\_HAS\_SUPPORT\_ENGINE\_CLK\_SEL\_REMOVE feature to support SoCs without CAN Engine Clock selection in FlexCAN module.
    - \* Added FlexCAN Serial Clock Operation to support i.MX SoCs.
- 2.1.0
  - Bug fixes:
    - \* Fixed wrong function name spelling: FLEXCAN\_XXX() -> FLEXCAN\_XXX());
    - \* Moved Freeze Enable/Disable setting from FLEXCAN\_Enter/ExitFreezeMode() to FLEXCAN\_Init();
    - \* Fixed wrong helper macro values.
  - Other changes:
    - \* Hid FLEXCAN\_Reset() to user.
    - \* Used NDEBUG macro to wrap FLEXCAN\_IsMbOccupied() function instead of DEBUG macro.
- 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 2.1.4.

- 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 2.1.5.

- 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 2.1.4.

- 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.

## **GPIO**

The current driver version is 2.2.1.

- 2.2.1:
  - API interface changes:
    - \* Refined naming of API while keep all original APIs by marking them as deprecated. Original API will be removed in next release. The main change is update API with prefix of \_PinXXX() and \_PortXXX.
- 2.1.1:
  - API interface changes:
    - \* Added API for the check attribute bytes.
- 2.1.0:
  - API interface changes:
    - \* Added "pins" or "pin" to some APIs' names.
    - \* Renamed "\_PinConfigure" to "GPIO\_PinInit".

## LLWU

The current LLWU driver version is 2.0.1.

- 2.0.1
  - Miscellaneous changes:
    - \* Updates for KL8x.
- 2.0.0
  - Initial version.

## LPI2C

The current LPI2C driver version is 2.1.5.

- 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. In the previous way the API uses scatter gather method, handle command transfer first, then handles the linked TCD which preset with the receive data transfer. The issue is that TX DMA request and RX DMA request are both enabled, when DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happens due to TX FIFO empty. This results the RX DMA transfer starts, without waiting on the expected RX DMA request. Fix 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 autostop 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 can not 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.

## LPTMR

The current LPTMR driver version is 2.0.1.

- 2.0.1
  - Driver update:
    - \* Updated the LPTMR driver to support 32-bit CNR and CMR registers in some devices.
- 2.0.0
  - Initial version.

## LPUART

The current LPUART driver version is 2.2.5.

- 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.

## PDB

The current PDB driver version is 2.0.1.

- 2.0.1
  - Changed PDB register base array to const.
- 2.0.0
  - Initial version.

## PIT

The current PIT driver version is 2.0.0.

- 2.0.0
  - Initial version.

## PMC

The current PMC driver version is 2.0.0.

- 2.0.0

- Initial version.

## **PORT**

The current PORT driver version is 2.0.2.

- 2.0.2
  - Miscellaneous changes:
    - \* Added feature guard macros in the driver.
- 2.0.1
  - Miscellaneous changes:
    - \* Added "const" in function parameter.
    - \* Updated some enumeration variables' names.

## **RCM**

The current RCM driver version is 2.0.1.

- 2.0.1
  - [KPSDK-10249] Fixed kRCM\_SourceSw bit shift issue.
- 2.0.0
  - Initial version.

## **RTC**

The current RTC driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **SAI**

The current SAI driver version is 2.1.4.

-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.

## **SIM**

The current SIM driver version is 2.1.0.

- 2.1.0
  - Added new APIs of SIM\_GetRfAddr() and SIM\_EnableSystickClock().
- 2.0.0
  - Initial version.

## **SMARTCARD**

The current SMARTCARD driver version is 2.2.0.

- 2.2.0
  - New features:
    - \* Updated to use RX/TX FIFO.
- 2.1.2
  - Provided time delay function which works in microseconds.
  - Bug fix:
    - \* Changed event to semaphore in RTOS driver (KPSDK-11634).
    - \* Added check if de-initialized variables are not null in SMARTCARD\_RTOS\_Deinit() (KPSDK-8788).

- \* Changed deactivation sequence in SMARTCARD\_PHY\_TDA8035\_Deactivate() to properly stop the clock (POSCR-35).
- \* Fixed timing issue with VSEL0/1 signals in smartcard TDA8035 driver (KPSDK-10160).
- 2.1.1
  - New features:
    - \* Added default phy interface selection into smartcard RTOS drivers (KPSDK-9063).
    - \* Replaced smartcard\_phy\_ncn8025 driver by smartcard\_phy\_tda8035.
  - Bug fix:
    - \* Fixed protocol timers activation sequences in smartcard\_emvsim and smartcard\_phy\_tda8035 drivers during emv11 pre-certification tests (KPSDK-9170, KPSDK-9556).
- 2.1.0
  - Initial version.

## SMC

The current SMC driver version is 2.0.3.

- 2.0.3
  - Added APIs SMC\_PreEnterStopModes, SMC\_PreEnterWaitModes, SMC\_PostExitWaitModes, and SMC\_PostExitStopModes.
- 2.0.2
  - Bug fix:
    - \* Added DSB before WFI, add ISB after WFI.
  - Miscellaneous changes:
    - \* Updated SMC\_SetPowerModeVlpw implementation.
- 2.0.1
  - Miscellaneous changes:
    - \* Updated for KL8x.
- 2.0.0
  - Initial version.

## TPM

The current TPM driver version is 2.0.2.

- 2.0.2
  - Bug fixes:
    - \* Fixed issues in functions TPM\_SetupPwm/TPM\_UpdateChnlEdgeLevelSelect /TPM\_SetupInputCapture/TPM\_SetupOutputCompare/TPM\_SetupDualEdgeCapture, wait acknowledgement when channel disabled.
- 2.0.1
  - Bug fixes:
    - \* Fix TPM\_UpdateChnlEdgeLevelSelect ACK wait issue.
    - \* Fix TPM\_SetupDualEdgeCapture can not set FILTER register issue.

- \* Fix TPM\_UpdateChnEdgeLevelSelect ACK wait issue.
- 2.0.0
  - Initial version.

## UART

The current UART driver version is 2.1.5.

- 2.1.5
  - Added hardware flow control function support.
  - Added idle line detected feature in UART\_TransferNonBlocking function. If an idle line is detected, a callback is 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 all interrupts are not 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 not be bigger than the RX FIFO size). Data is then received and a callback is triggered when data receive ends.
- 2.1.4
  - Changed parameter type in UART\_RTOS\_Init() struct rtos\_uart\_config -> uart\_rtos\_config\_t.
  - Bug fixed:
    - \* Disabled UART receive interrupt instead of disable all NVIC when read data from ring buffer. Because with 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 interrupt.
- 2.1.3
  - Added RX framing error and parity error status check when use interrupt transfer.
- 2.1.2
  - Fixed baud rate fine adjust bug to make the computed baud rate more accurately.
- 2.1.1
  - Removed needless check of event flags and assert in UART\_RTOS\_Receive.
  - Waited always for RX event flag in UART\_RTOS\_Receive.
- 2.1.0
  - Added transactional API.
- 2.0.0
  - Initial version.

## WDOG

The current WDOG driver version is 2.0.0.

- 2.0.0
  - Initial version.

## **CLOCK**

The current CLOCK driver version is 2.2.1.

- 2.2.1
  - Bug fixes:
    - \* Fixed the issue where MCG could not switch to FEE/FBE/PBE modes when OSCERCLK clock not enabled.
- 2.2.0
  - New features:
    - \* [KPSDK-9157] Update CLOCK\_SetFeiMode/CLOCK\_SetFbiMode/CLOCK\_BootToFeiMode() to support set MCG\_C4[DMX32]=1 in FEI/FBI modes.
  - Bug fixes:
    - \* Updated IP\_CLOCKS array, removed unused gates and add missing gates.
- 2.1.0
  - Other changes:
    - \* Merge fsl\_mcg and fsl\_osc into fsl\_clock.
- 2.0.0
  - Initial version.

## 2 Middleware Change Log

### DMA\_MANAGER

The current DMA\_MANAGER driver version is 2.1.0.

- 2.1.0
  - Updated DMA manager interface to support dynamic configuration of the managed area. This is used for a platform with multiple cores.
- 2.0.0
  - Initial version.

### EMVL1 for MCUXpresso SDK

The current driver version is 2.1.0.

- 2.1.0
  - Added abort transfer functionality.
- 2.0.2
  - Re-implemented function for sending commands in T=0.
  - Bug fixes:
    - \* Fixed wrong size of response in T=0 (KPSDK-11248).
    - \* Fixed problem with command cases 3 in T=1, expected wrong length of response (KPSDK-11335).
    - \* Fixed wrong length of response in T=1 (KPSDK-11868).
    - \* Fixed usage application buffer for data payload and overhead associated with T=1 protocol (KPSDK-11336).
- 2.0.1
  - Bug fixes:
    - \* Fixed low level driver protocol timers failures during emvl1 pre-certification tests (KPSDK-9556).
    - \* Fixed improper T0 commands response receiving (commands case2, case3 & case4 affected) what causes long commands responses (KPSDK-8707).
- 2.0.0
  - Initial version.

### FatFs for MCUXpresso SDK

The current version is FatFs R0.13a\_rev0.

- R0.13a\_rev0
  - Upgraded to version 0.13a. Added patch ff\_13a\_p1.diff.
- R0.12c\_rev1

- Added 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.

## **SDMMC**

The current driver version is 2.2.4.

- 2.2.4
  - Bug fixes:
    - \* Fixed DDR mode data sequence miss issue, which is caused by NIBBLE\_POS.
  - New features:
    - \* Increased g\_sdmmc 512byte to improve the performance when application use a non-word align data buffer address.
    - \* Used OCR access mode bits to determine the mmccard high capacity flag.
    - \* Enabled auto cmd12 for SD read/write.
    - \* Disabled DDR mode frequency multiply by 2.
- 2.2.3
  - Bug fixes:
    - \* Added reponse check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.2
  - Moved set card detect priority operation before enable IRQ.
- 2.2.1
  - New features:
    - \* Improved MMC Boot feature.
    - \* Keep SD\_Init/SDIO\_Init function for forward compatibility.
- 2.2.0
  - New features:
    - \* Separated the SD/MMC/SDIO init API to xxx\_CardInit/xxx\_HostInit.
    - \* Allowed user register card detect callback, select card detect type, and determine the card detect timeout value.
    - \* Allowed user register the power on/off function, and determine the power on/off delay time.
    - \* SD\_Init/SDIO\_Init will be deprecated in the next version.

- \* Added write complete wait operation for MMC\_Write to fix command timeout issue.
- 2.1.6
  - Enhanced SD IO default driver strength.
- 2.1.5
  - Fixed coverity issue.
  - Fixed SD v1.x card write fail issue. It was caused by the block length set error.
  - Improved SDIO card init sequence and add retry option for SDIO\_SwitchToHighSpeed function.
- 2.1.4
  - Miscellaneous:
    - \* Added Host reset function for card re-initialization.
    - \* Added Go\_Idle function for SDIO card.
    - \* Added Host\_ErrorRecovery function for host error recovery procedure.
    - \* Added cache maintain operation
    - \* Added HOST\_CARD\_INSERT\_CD\_LEVEL to improve compatibility.
  - Bug fixes:
    - \* Fixed card cannot detect dynamically.
- 2.1.3
  - Bug fixes:
    - \* Non high-speed sdc card init fail at switch to high speed.
  - Miscellaneous:
    - \* Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function.
    - \* Added strobe dll for mmc HS400 mode.
    - \* Added Delay for SDCard power up.
- 2.1.2
  - New features:
    - \* Added fsl\_host.h to provide prototype to adapt different controller IPs(SDHC/SDIF).
    - \* Added adaptor code in SDMMC/Port folder to adapt different host controller IPs with different transfer modes(interrupt/polling/freertos). Application includes a different adaptor code to make application more simple.
    - \* Adaptor code provides HOST\_Init/HOST\_Deinit/CardInsertDetect. APIs to do host controller initialize and transfer function configuration. SDMMC card stack uses adaptor code inside stack to wait card insert and configure host when calling card init APIs (SD\_Init/MMC\_Init/SDIO\_Init).
    - \* This change requires the user to include host adaptor code into the application. If not changed, link errors saying it cannot find the definition of HOST\_Init/HOST\_Deinit/CardInsertDetect appear.
  - New features: Improved SDMMC to support SD v3.0 and emmc v5.0.
  - Bug fixes:
    - \* Fixed wrong comparison between count and length in MMC\_ReadBlocks/MMC\_WriteBlocks.
- 2.1.1
  - Bug fixes:
    - \* Fixed the block range boundary error when transferring data to MMC card.

- \* Fixed the bit mask error in the SD card switch to high speed function.
- Other changes:
  - \* Added error code to indicate that SDHC ADMA1 transfer type is not supported yet.
  - \* Optimized the SD card initialization function.
- 2.1.0
  - Bug fixes:
    - \* Changed the callback mechanism when sending a command.
    - \* Fixed the performance low issue when transferring data.
  - Other changes:
    - \* Changed the name of some error codes returned by internal function.
    - \* Merged all host related attributes to one structure.
    - \* Optimize the function of setting maximum data bus width for MMC card.

## USB stack for MCUXpresso SDK

The current version of USB stack is 2.0.1.

- 2.0.1
  - Bug fixes:
    - \* Fixed some USB issues.
    - \* Change the audio codec interfaces.
- 2.0.0
  - New features:
    - \* PTN5110N support.
  - Bug fixes:
    - \* 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 fixes:
    - \* 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 fixes:

- Code cleaned 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 fixes: -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

## 3 RTOS Change Log

### FreeRTOS for MCUXpresso SDK

The current version is FreeRTOS 9.0.0. Original package is available at [freertos.org](http://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`.
- 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 `configFRRTOS_MEMORY_SCHEME` macros.
- 9.0.0\_rev1
  - New features:
    - \* Enabled `-fno` 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 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).

NXP, the NXP logo, Freescale, the Freescale logo, Kinetis, Processor Expert are trademarks of NXP B.V. Tower is a trademark of NXP. All other product or service names are the property of their respective owners. Arm and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved.

© 2018 NXP B.V.

