Document Number: MCUXSDKKL27ZRN

Rev. 0, 05/2018

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z

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 FreeRTOSTM 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.

Contents

1	Overview	l
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	4
8	Known issues	6

2 MCUXpresso SDK



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)[®] 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
FRDM-KL27Z	MKL27Z32VDA4, MKL27Z32VLH4, MKL27Z32VFM4, MKL27Z32VMP4, MKL27Z32VFT4, MKL27Z64VDA4, MKL27Z64VFT4, MKL27Z64VFM4, MKL27Z64VFH4, MKL17Z32VDA4, MKL17Z32VFT4, MKL17Z32VLH4, MKL17Z32VFM4, MKL17Z32VMP4, MKL17Z64VDA4, MKL17Z64VPM4, MKL17Z64VFM4, MKL17Z64VFT4, MKL17Z64VFM4, MKL17Z64VFT4, MKL17Z64VFM4

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</install_dir>

Table continues on the next page...

Table 2. Release contents (continued)

<install_dir>/boards/<board_name>/demo_apps</board_name></install_dir>
<install_dir>/boards/<board_name>/driver_examples</board_name></install_dir>
<install_dir>/boards/<board_name>/rtos_examples</board_name></install_dir>
<install_dir>/boards/<board_name>/multiprocessor_examples</board_name></install_dir>
<install_dir>/docs</install_dir>
<install_dir>/middleware</install_dir>
<install_dir>/middleware/dma_manager</install_dir>
<install_dir>/middleware/fatfs</install_dir>
<install_dir>/middleware/multicore</install_dir>
<install_dir>/middleware/sdmmc</install_dir>
<install_dir>/devices/<device_name></device_name></install_dir>
<install_dir>/CMSIS</install_dir>
<install_dir>/devices/<device_name>/drivers</device_name></install_dir>
<install_dir>/devices/<device_name>/utilities</device_name></install_dir>
<install_dir>/rtos</install_dir>
<install_dir>/tools</install_dir>

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.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 05/2018

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 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.

6.2.2 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.

7 MISRA compliance

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

Table 3. MISRA exceptions

Exception Rules	Description
Directive 4.4	Sections of code should not be commented out.
	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.

Table continues on the next page...

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 05/2018

Table 3. MISRA exceptions (continued)

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.
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 linage 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.

Table continues on the next page...

Known issues

Table 3. MISRA exceptions (continued)

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 II operator shall not contain persistent side effects.
Rule 14.2	A for loop shall be well formed.
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 17.7	The feature of <stdarg.h> shall not be used.</stdarg.h>
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.

8.4 FreeRTOS issue

When generating a new FreeRTOS project with New Project Wizard tool, the user should deassert the micro option "configUSE_PORT_OPTIMISED_TASK_SELECTION" in the "FreeRTOSConfig.h" file while it is not used in the template project for some Cortex-M0+ devices, or it might not pass the compiling.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 05/2018

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z

Change Logs

Contents

Driver	Change Log	1
	ADC16	1
	CMP	1
	COP	1
	CRC	1
	DMA	1
	DMAMUX	2
	FLASH	2
	FLEXIO	3
	FLEXIO_UART	3
	FLEXIO_I2C	4
	FLEXIO_SPI	5
	FLEXIO_I2S	5
	FLEXIO_MCU_LCD	6
	FLEXIO_CAMERA	6
	GPIO	7
	12C	7
	LLWU	8
	LPTMR	8
	LPUART	9
	PIT	10

Contents	Page
Title	Number
PMC	10
PORT	10
RCM	10
RTC	10
SIM	11
SMC	11
SPI	11
TPM	12
UART	12
VREF	13
CLOCK	13
Middleware Change Log	14
DMA_MANAGER	14
EMVL1 for MCUXpresso SDK	14
FatFs for MCUXpresso SDK	14
SDMMC	15
USB stack for MCUXpresso SDK	17
RTOS Change Log	21
FreeRTOS for MCUXpresso SDK	21

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.

COP

The current COP 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.

DMA

The current DMA driver version is 2.0.1.

- 2.0.1
 - Bug fix:
 - * Fixed the DMA driver build fail due to MISRA C 2004 rule 12.5 by adding parenthesis.
- 2.0.0
 - Initial version.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

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.

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 FTFL 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 flexnvm.
 - * Improved the implementation of execute-in-ram functions.
- 2.0.0

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- 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 enableIn-Doze = 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:

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

4

- * Changed the API name FLEXIO UART StopRingBuffer to FLEXIO UART Transfer-StopRingBuffer 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 12C

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->baud-Rate_Bps) / 2 - 1" exceeds 0xFFU.
- 2.1.2
 - Fixed the FLEXIO I2C issue where the master cannot receive data from I2C slave in high
 - Fixed the FLEXIO I2C issue where the master cannot receive NAK when master sends nonexistent 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_MasterTransfer-CreateHandle issue. This leads the s_flexioHandle/s_flexioIsr/s_flexioType variable written. Then, if calling FLEXIO_I2C_MasterTransferBlocking API multiple times, the s_flexio-Handle/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

NXP Semiconductors

- Bug fixes:
 - * Implemented the FLEXIO_I2C_MasterTransferBlocking API which defined in header file

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

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 12S

The current FLEXIO I2S driver version is 2.1.4.

- 2.1.4
 - Unify component full name to FLEXIO I2S(DMA/EDMA) Driver

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- 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

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- 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".

I₂C

The current I2C driver version is 2.0.5.

- 2.0.5
 - Improvements:
 - * Added I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.4
 - Bug fixes:
 - * Added proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Only supports write only or write+read with no start flag, does not support read only with no start flag.
- 2.0.3
 - Bug fixes:
 - * Removed enableHighDrive member in the master/slave configuration structure because the operation to HDRS bit is useless, user needs to use DSE bit in port register to configure the high drive capability.
 - * Added reset registers operation in I2C_MasterInit and I2C_SlaveInit APIs. Fixed issue where I2C could not switch between master and slave mode.
 - * Improved slave IRQ handler to handle the corner case that stop flag and address match

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

flag come synchronously.

- 2.0.2
 - Bug fixes:
 - * Fixed issue in master receive and slave transmit mode with no stop flag. The master could not succeed to start next transfer because the master could not send out re-start signal.
 - * Fixed data transfer out of order issue due to memory barrier
 - * Added hold time configuration for slave. By leaving the SCL divider and MULT reset values when configure to slave mode, the setup and hold time of the slave is then reduced outside of spec for lower baudrates. This can cause intermittent arbitration loss on the master side.
 - New features:
 - * Added address nak event for master.
 - * Added general call event for slave.
- 2.0.1
 - New features:
 - * Added double buffer enable configuration for Socs which have the DFEN bit in S2 register.
 - * Added flexible transmit/receive buffer size support in I2C_SlaveHandleIRQ.
 - * Added start flag clear, address match, and release bus operation in I2C_SlaveWrite/Read-Blocking API.
 - Bug fix:
 - * Changed the kI2C_SlaveRepeatedStartEvent to kI2C_SlaveStartEvent.

LLWU

The current LLWU driver version is 2.0.1.

- 2.0.1
 - Miscellaneous changes:
 - * Updates for KL8x.
- 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.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

9

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_configt.
- 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 seprarte 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 2.0.0.

- 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.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

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.

SMC

The current SMC driver version is 2.0.3.

- 2.0.3
 - Added APIs SMC_PreEnterStopModes, SMC_PreEnterWaitModes, SMC_PostExitWait-Modes, 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.

SPI

The current SPI driver version is 2.0.4.

- 2.0.4
 - New feature:
 - * Supports 3-wire mode for SPI driver. Added new API SPI_SetPinMode() to control the transfer direction of the single wire. For master instance, MOSI is selected as I/O pin. For slave instance, MISO is selected as I/O pin.
 - * Added dummy data setup API to allow users to configure the dummy data to be transferred.
- 2.0.3
 - Bug fix:
 - * Fixed the potential interrupt race condition in high baudrate when call API SPI_Master-TransferNonBlocking.
- 2.0.2
 - New feature:
 - * Allows user to set the transfer size for SPI_TransferNoBlocking non-integer times of wa-

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

termark.

- * Allows user to define the dummy data, users only need to define the macro SPI_DUMM-YDATA in applications.
- 2.0.1
 - Bug fix:
 - * Fixed SPI_Enable function parameter error.
 - * Set the s_dummy variable as static variable in fsl_spi_dma.c.
 - Optimazation:
 - * Optimized the code size while not use transactional API.
 - * Improved performance in polling method.
 - * Added #ifndef/#endif to allow user to change the default tx value at compile time.
- 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_UpdateChnIEdgeLevelSelect 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).

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

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.

VREF

The current VREF driver version is 2.1.0.

- 2.1.0
 - Added new functions:
 - * Supported L5K board: add VREF_SetTrim2V1Val() and VREF_GetTrim2V1Val() functions to supply 2V1 output mode.
- 2.0.0
 - Initial version.

CLOCK

The current CLOCK driver version is 2.1.1.

- 2.1.1
 - Bug fixes:
 - * Updated IP_CLOCKS array, removed unused gates and added missing gates.
- 2.1.0
 - Other changes:
 - * Merged fsl mcglite and fsl osc into fsl clock.
- 2.0.0
 - Initial version.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

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 Fix:
 - * Fixed wrong size of response in T=0 (KPSDK-11248).
 - * Fixed problem with command cases 3 in T=1, expected wrong length of response (KPS-DK-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 fix:
 - * Fixed low level driver protocol timers failures during emvl1 pre-certification tests (KPS-DK-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

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- 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 fix:
 - * 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 fix:
 - * 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.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- * 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 fix:
 - * Fixed card cannot detect dynamically.
- 2.1.3
 - Bug fix:
 - * Non high-speed sdcard 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 fix:
 - * Fixed wrong comparison between count and length in MMC_ReadBlocks/MMC_Write-Blocks.
- 2.1.1
 - Bug fix:
 - * Fixed the block range boundary error when transferring data to MMC card.

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

17

- * 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 fix:
 - * Change the callback mechanism when sending a command.
 - * Fix 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 fix:
 - * fixed some USB issues.
 - * 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 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 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

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- * 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

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

- * 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

MCUXpresso SDK Release Notes Supporting FRDM-KL27Z, Rev. 0, 5/2018

21

3 RTOS Change Log

FreeRTOS for MCUXpresso SDK

The current version is FreeRTOS 9.0.0. Original package is available at 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_SU-PPORT 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 configFRTOS_MEMORY SCHEME macros.
- 9.0.0 rev1
 - New features:
 - * Enabled -flto 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 apropriate file fsl_tickless_lptmr.c.
 - * Removed configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of **attribute**((weak)) is preffered solution. Not same as week!
- 8.2.3
 - New features:
 - * Tickles 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

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP 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.

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 μ 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 MCUXSDKKL27ZRN Revision 0, 05/2018



