

MCUXpresso SDK Release Notes Supporting TWR-KE18F and HVP-KE18F

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)® 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 |
|---------------------|--|
| TWR-KE18F, HVP-KE18 | MKE14F256VLL16, MKE14F512VLL16, MKE14F256VLH16, MKE14F512VLH16, MKE16F256VLL16, MKE16F512VLL16, MKE16F256VLH16, MKE16F512VLH16, MKE18F256VLL16, MKE18F512VLL16 , MKE18F256VLH16, MKE18F512VLH16 |

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 |

Table continues on the next page...

Table 2. Release contents (continued)

| | |
|---|---|
| RTOS examples | <install_dir>/boards/<board_name>/rtos_examples |
| 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 |
| | HYUNDAI HY-HB608 |
| USB flash drive | ADATA C008 32 GB |
| | ADATA S102 8 G |
| | ADATA S102 16 G |
| | Verbatim STORE N GO USB Device 8 G |

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/adapter | 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 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. |
| 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.

MCUXpresso SDK Release Notes Supporting TWR-KE18F and HVP-KE18F

Change Logs

Contents

| | |
|---------------------------------|-----------|
| Driver Change Log | 1 |
| ACMP | 1 |
| ADC12 | 1 |
| CRC | 1 |
| DAC32 | 1 |
| DMAMUX | 2 |
| EDMA | 2 |
| EWM | 3 |
| FLASH | 3 |
| FLEXCAN | 4 |
| FLEXIO | 4 |
| FLEXIO_UART | 5 |
| FLEXIO_I2C | 6 |
| FLEXIO_SPI | 6 |
| FLEXIO_I2S | 7 |
| FLEXIO_MCU_LCD | 8 |
| FLEXIO_CAMERA | 8 |
| FTM | 8 |
| GPIO | 9 |
| LMEM | 9 |
| LPI2C | 10 |

Contents

| | Page Number |
|--|----------------|
| LPIT | 11 |
| LPSPi | 11 |
| LPTMR | 11 |
| LPUART | 12 |
| PDB | 13 |
| PMC | 13 |
| PORT | 13 |
| PWT | 13 |
| RCM | 13 |
| RTC | 14 |
| SIM | 14 |
| SMC | 14 |
| SYSMPU | 14 |
| TRGMUX | 15 |
| WDOG32 | 15 |
| CLOCK | 15 |
| Middleware Change Log | 17 |
| DMA_MANAGER | 17 |
| FatFs for MCUXpresso SDK | 17 |
| RTOS Change Log | 18 |
| FreeRTOS for MCUXpresso SDK | 18 |

1 Driver Change Log

ACMP

The current ACMP driver version is 2.0.3.

- 2.0.3
 - Added feature functions for different power domain's usage (1.8 V and 3 V). These functions are first enabled in ULP1. They are about:
 - * ACMP_EnableLinkToDAC()
 - * ACMP_SetDiscreteModeConfig()
 - * ACMP_GetDefaultDiscreteModeConfig()
- 2.0.2
 - Coding style changes:
 - * Changed coding style of peripheral base address from "s_acmpBases" to "s_acmpBase";
- 2.0.1
 - Bug fix:
 - * Fixed bug regarding the function "ACMP_SetRoundRobinConfig". It will not continue execution but returns directly after disabling round robin mode;

ADC12

The current ADC12 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.

DAC32

The current DAC32 driver version is 2.0.1.

- 2.0.1
 - Bug fix:

- * Moved the default DAC32_Enable(..., true) from DAC32_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.

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 FTFLL 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
 - 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_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_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->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 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.

FTM

The current FTM driver version is 2.0.4.

- 2.0.4
 - Features:
 - * Added to enable DMA transfer with new API:
 - FTM_EnableDmaTransfer()
- 2.0.3
 - Bug fixes:
 - * Updated the FTM driver to enable fault input after configuring polarity.
- 2.0.2
 - Features:
 - * Added support to Quad Decoder feature with new APIs:
 - FTM_GetQuadDecoderFlags()
 - FTM_SetQuadDecoderModuloValue()
 - FTM_GetQuadDecoderCounterValue()
 - FTM_ClearQuadDecoderCounterValue()
- 2.0.1
 - Bug fixes:
 - * Updated the FTM driver to fix write to ELSA and ELSB bits.
 - * FTM combine mode: set the COMBINE bit before writing to CnV register.
- 2.0.0
 - Initial version.

GPIO

The current 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".

LMEM

The current LMEM driver version is 2.1.0.

- 2.1.0
 - Removed the write buffer enable from the cache enable API.

- Added Enable write buffer APIs.
- 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.

LPIT

The current LPIT driver version is 2.0.0.

- 2.0.0
 - Initial version.

LPSPI

The current LPSPI driver version is 2.0.2.

- 2.0.2
 - New feature:
 - * Added dummy data setup API to allow users to configure the dummy data to be transferred.
 - * Enabled the 3-wire mode, SIN and SOUT pins can be configured as input/output pin.
- 2.0.1
 - Bug fix:
 - * The clock source should divided by PRESCALE setting in LPSPI_MasterSetDelayTimes function.
 - * Fixed the bug that LPSPI_MasterTransferBlocking function would hang in some corner cases.
 - Optimization:
 - * Added #ifndef/#endif to allow user to change the default TX value at compile time.
- 2.0.0
 - Initial version.

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.

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.

PWT

The current PWT driver version is 2.0.0.

- 2.0.0
 - Initial version.

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.

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_PostExitWaitModes, and SMC_PostExitStopModes.
- 2.0.2
 - Bug fix:
 - * Added DSB before WFI, add ISB after WFI.
 - Miscellaneous changes:
 - * Updated SMC_SetPowerModeVlps implementation.
- 2.0.1
 - Miscellaneous changes:
 - * Updated for KL8x.
- 2.0.0
 - Initial version.

SYSMPU

The current SYSMPU driver version is 2.2.1.

- 2.2.1
 - Fixed MISRA issue.
- 2.2.0
 - Renamed MPU to SYSMPU.
 - Changed macro definition for slave number and fix the get error status calculation.
- 2.1.1
 - Added the feature file macro definition limitation for the MPU_SetRegionRwMasterAccess-

- Rights().
- 2.1.0
 - API changes:
 - * Changed the mpu_region_num_t and mpu_master_t to uint32_t.
 - * Changed the mpu_low_masters_access_rights_t, mpu_high_masters_access_rights_t to mpu_rwxrights_master_access_control_t, mpu_rwrights_master_access_control_t.
 - * Changed the MPU_SetRegionLowMasterAccessRights(), MPU_SetRegionHighMasterAccessRights() to MPU_SetRegionRwxMasterAccessRights(), MPU_SetRegionRwMasterAccessRights().
- 2.0.0
 - Initial version.

TRGMUX

The current TRGMUX driver version is 2.0.0.

- 2.0.0
 - Initial version.

WDOG32

The current WDOG32 driver version is 2.0.1.

- 2.0.1
 - Bug fixes:
 - * WDOG must be configured within it's configuration time period
 - Added WDOG32_Init API to quick access section.
 - Defined register variable in WDOG32_Init API.
- 2.0.0
 - Initial version.

CLOCK

The current CLOCK driver version is 2.1.2.

- 2.1.2
 - Bug fixes:
 - * Fixed issue where OSC32 cannot use EXTAL32 clock source.
- 2.1.1
 - Improvements:
 - * Changed reserved bit fields in _scg_sys_clk_config struct into unnamed bit fields.
- 2.1.0
 - Other changes:
 - * Merged fsl_scg 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.

FatFs for MCUXpresso SDK

Current version is FatFs R0.13a_rev0.

- R0.13a_rev0
 - Upgraded to version 0.13a. Added patch ff_13a_p1.diff.
- R0.12c_rev1
 - Add nand disk support.
- R0.12c_rev0
 - Upgraded to version 0.12c and applied patches ff_12c_p1.diff and ff_12c_p2.diff.
- R0.12b_rev0
 - Upgraded to version 0.12b.
- R0.11a
 - Added glue functions for low-level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c.
 - Added RTOS wrappers to make FatFs thread safe. Modified syscall.c.
 - Renamed ffconf.h to ffconf_template.h. Each application should contain its own ffconf.h.
 - Included ffconf.h into diskio.c to enable the selection of physical disk from ffconf.h by macro definition.
 - Conditional compilation of physical disk interfaces in diskio.c.

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_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 configFRTOS_MEMORY_SCHEME macros.
- 9.0.0_rev1
 - New features:
 - * Enabled -fcto 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**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 MCUXSDKKE18RN
Revision 0, 05/2018

