NXP Semiconductors
User's Guide

Document Number: MCUXSDKKV31RN
Rev. 0, 06/2019

MCUXpresso SDK Release Notes Supporting TWR-KV31F120M, FRDM-KV31F, and HVP-KV31F120M

Contents

Chapter 1
Overview

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

For more details about MCUXpresso SDK, see the MCUXpresso SDK homepage MCUXpresso-SDK: Software Development Kit.

NOTE

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

Chapter 2
MCUXpresso SDK

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

NOTE

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

Chapter 3
Development tools

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

- IAR Embedded Workbench for Arm version 8.32.3
- MDK-Arm Microcontroller Development Kit (Keil)® 5.27
- Makefiles support with GCC revision 8-2018-q4-major GCC8 from Arm Embedded
- MCUXpresso IDE v11.0.0

Chapter 4
Supported development systems

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

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
TWR-KV31F120M, FRDM-KV31F, HVP-KV31F120M	**MKV31F512VLL12**, MKV31F512VLH12, MKV30F128VFM10, MKV30F128VLF10, MKV30F128VLH10, MKV30F64VFM10, MKV30F64VLF10, MKV30F64VLH10, MKV31F128VLH10, MKV31F128VLL10, MKV31F256VLH12, MKV31F256VLL12

Chapter 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
TinyCBOR	<install_dir>/rtos/amazon-freertos/lib/third_party/tinycbor
Demo applications	<install_dir>/boards/<board_name>/demo_apps
Driver examples	<install_dir>/boards/<board_name>/driver_examples
Cortex Microcontroller Software Interface Standard (CMSIS) driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
FatFS examples	<install_dir>/boards/<board_name>/fatfs_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
FatFS stack	<install_dir>/middleware/fatfs
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
CMSIS Arm Cortex®-M header files, DSP library source	<install_dir>/CMSIS
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools
segger_systemview	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_segger_sysview
percepio_snapshot	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_percepio_snapshot

Chapter 6
MCUXpresso SDK release package

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

6.1 Device support

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

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

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

6.1.1 Board support

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

6.1.2 Demo applications and other examples

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

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

6.2 Middleware

6.2.1 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.

Chapter 7
MISRA compliance

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

Table 3. MISRA exceptions

Exception Rules	Description
Rule 5.1	External identifiers shall be distinct.
Rule 5.4	Macro identifiers shall be distinct.
Rule 21.1	#define and #undef shall not be used on a reserved identifieror reserved macro name.
Rule 21.2	A reserved identifier or macro name shall not be declared.
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	Typedefs that indicate size and signedness should be used in place of the basic numerical types.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden.
Directive 4.9	A function should be used in preference to a function-like macro where they are interchangeable.
Directive 4.13	Functions which are designed to provide operations on a resource should be called in an appropriate sequence.
Rule 1.2	Language extensions should not be used.
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 2.6	A function should not contain unused label declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 4.2	Trigraphs should not be used.
Rule 5.9	Identifiers that define objects or functions with internal linkage should be unique.
Rule 8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.
Rule 8.9	An object should be defined at block scope if its identifier only appears in a single function.
Rule 8.11	When an array with external linkage is declared, its size should be explicitly specified.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential 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 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.3	The comma operator should not be used.
Rule 12.4	Evaluation of constant expressions should not lead to unsigned integer wrap-around.
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 15.4	There should be no more than one break or go to statement used to terminate any iteration statement.
Rule 17.5	The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements.
Rule 17.8	A function parameter should not be modified.
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 21.12	The exception handling features of <fenv.h> should not be used. .

Chapter 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 Program flash issue

The last 8 KB flash segment is restricted to execute only by default, and users cannot break the restriction. This means that there is 512 KB flash in KV31, but only the first 504 KB flash is accessible as normal flash for the users.

8.5 Create new project without board template

The following components should be selected at the same time when creating a new project without using a board template, including serial_manager, serial_manager_uart, debug_console, and one UART adapter (lpuart_adapter for LPUART IP, uart_adapter for UART IP, lpsci_adapter for LPSCI IP, etc).

8.6 New Project Wizard compile failure

The following components request the user to manually select other components that they depend on to pass the compile. These components depend on several components, and the New Project Wizard (NPW) is not able to decide which one is needed by the user.

NOTE

"xxx"means core variants like cm0plus, cm33, cm4, cm33_nodsp.

Components: Assert, assert_cm0plus, assert_xxx, assert_lite, baremetal, button, codec_i2c, codec_i2c_xxx, debug_console, debug_console_xxx, debug_console_lite, dialog7212, led, misc_utilities, panic, serial_manager, serial_manager_xxx,

serial_manager_swo, serial_manager_swo_xxx, serial_manager_uart, serial_manager_uart_xxx, serial_manager_usb_cdc, serial_manager_usb_cdc_xxx, sgtl_adapter, sgtl5000, shell, shell_xxx, timer_manager, wm8904, wm8904_xxx, wm8904_adapter, wm8904_adapter_xxx, wm8960, wm8960_adapter, xip_device.

MCUXpresso SDK Release Notes Supporting TWR-KV31F120M, FRDM-KV31F, and HVP-KV31F120M

Change Logs

Contents

Contents

1 Driver Change Log

ADC16

The current ADC16 driver version is 2.0.2.

- 2.0.2
  - Used conversion control feature macro instead of that in IO map.
- 2.0.1
  - Bug fix:
    - Fixed MISRA-2012 rules.
      - Rule 16.4, 10.1, 13.2, 14.4, 17.7.
- 2.0.0
  - Initial version

CMP

The current CMP driver version is 2.0.1.

- 2.0.1
  - Bug fix:
    - Fixed MISRA-2012 rules.
      - Rule 14.4, rule 10.3, rule 10.1, rule 10.4, rule 17.7.
- 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 2.0.3.

- 2.0.3
  - Fix the issue for MISRA-2012 check.
    - Fixed rule 10.4, rule 10.3.
- 2.0.2
  - New feature:
    - Added an always-on enable feature to a DMA channel for ULP1 DMAMUX support.
- 2.0.1
  - Bug fix:
    - Fixed build warning while setting the DMA request source in DMAMUX_SetSource-Change issue by changing the type of the parameter source from uint8_t to uint32_t.
- 2.0.0
  - Initial version.

DSPI

The current DSPI driver version is 2.2.1.

- 2.2.1
  - Bug fix:
    - Fixed the bug for double execution of transfer complete callback in master interrupt transfer mode. In the interrupt routine, the DSPI interrupt may meet the situation of the interrupt pending by itself while receiving the last frame, adding check the transfer state to execute the callback function.
    - Fixed the wrong logic in DSPI_SetFifoEnable().
  - MISRA C-2012 issue fixed.
    - Fixed rule contain: rule-12.1, rule-17.7, rule-16.4, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.1, rule-10.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3, rule-8.5.
- 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_MasterTransfer-CreateHandleEDMA function like below: DSPI_MasterTransferCreateHandleEDMA(E-XAMPLE_DSPI_MASTER_BASEADDR, &g_dspi_edma_m_handle, DSPI_Master-UserCallback, &userData, &dspiEdmaMasterRxRegToRxDataHandle, NULL, &dspi-

EdmaMasterIntermediaryToTxRegHandle);
    - 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 2.1.8.

- 2.1.8
  - Bug fix:
    - Fixed wrong channel preemption base address used in EDMA_SetChannelPreemptionConfig api that will cause channel preemption register cannot configure correctly.
- 2.1.7
  - Bug fix:
    - Fixed wrong transfer size setting
      - Add 8 bytes transfer configuration and feature for RT series
      - Add feature to support 16 bytes transfer for Kinetis
    - Fixed the issue that EDMA_HandleIRQ will go to incorrect branch When TCD is not used and callback function is not registered.
- 2.1.6

  - Bug fix:
    - Fixed KW3X MISRA Issue.
      - Rule 14.4, 10.8, 10.4, 10.7, 10.1, 10.3, 13.5, 13.2.
  - Improvements:
    - Clear IRQ handler that not available for specific platform with macro FSL_FEATURE_-EDMA_MODULE_CHANNEL_IRQ_ENTRY_SHARED_OFFSET.
- 2.1.5
  - Improvements:
    - Improve EDMA IRQ handler to support half interrupt feature.
- 2.1.4
  - Bug fix:
    - Clear enabled request, status during EDMA_Init for the case that EDMA is halted before reinitialization.
- 2.1.3
  - Bug fix:
    - Add clear DONE bit in IRQ handler to avoid overwrite TCD issue.
    - Optimize above solution for the case that transfer request occurs in callback.
- 2.1.2
  - Improvements:
    - Added interface to get next TCD address.
    - Added interface to get the unused TCD number.
- 2.1.1
  - Improvements:
    - Added documentation for eDMA data flow when scatter/gather is implemented for the EDMA_HandleIRQ API.
    - Updated and corrected some related comments in the EDMA_HandleIRQ API and edma-_handle_t struct.
- 2.1.0
  - Improvements:
    - Changed the EDMA_GetRemainingBytes API into EDMA_GetRemainingMajorLoop-Count due to eDMA IP limitation (see API comments/note for further details).
- 2.0.5
  - Improvements:
    - Added pubweak DriverIRQHandler for K32H844P (16 channels shared).
- 2.0.4
  - Improvements:
    - Added support for SoCs with multiple eDMA instances.
    - Added pubweak DriverIRQHandler for KL28T DMA1 and MCIMX7U5_M4.
- 2.0.3
  - Bug fix:
    - Fixed the incorrect pubweak IRQHandler name issue, which causes re-definition build errors when client sets his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler.
- 2.0.2
  - Bug fix:

        - Fixed incorrect minorLoopBytes type definition in _edma_transfer_config struct, and defined minorLoopBytes as uint32_t instead of uint16_t.
- 2.0.1
  - Bug fix:
    - Fixed the eDMA callback issue (which did not check valid status) in EDMA_HandleIRQ API.
- 2.0.0
  - Initial version.

EWM

The current EWM driver version is 2.0.1.

- 2.0.1
  - Fixed EWM_Deinit hardfault issue.
- 2.0.0
  - Initial version.

FLASH

The current FLASH driver version is 3.0.0.

- 3.0.0
  - Improvements:
    - Reorganized FTFx flash driver source file.
    - Extracted flash cache driver from FTFx driver.
    - Extracted FLEXNVM flash driver from FTFx driver.
- 2.3.1
  - Bug fix:
    - 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 structure.
- 2.2.0
  - New features:
    - Supported FTFL device in FLASH_Swap API.
    - Supported various PFLASH start addresses.
    - Added support for KV58 in cache clear function.
    - Added support for device with secondary flash (KW40).

  - Bug fix:
    - 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.

FTM

The current FTM driver version is 2.1.1.

- 2.1.1
  - Fixed Coverity integer handing issue where the right operand of a left bit shift statement should not be a negative value. This appears in FTM_SetReloadPoints().
- 2.1.0
  - New feature:
    - Added a new API FTM_SetupPwmMode() to allow the user to set the channel match value in units of timer ticks. New configure structure called ftm_chnl_pwm_config_param_t was added to configure the channel's PWM parameters. This API is similar with FTM_SetupPwm() API, but the new API does not set the timer period (MOD value), it is useful for users to set the PWM parameters without changing the timer period.
  - Bug fixes:
    - Added feature macro to enable/disable the external trigger source configuration.
- 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 2.3.2.

- 2.3.2
  - Fixed the issue for MISRA-2012 check.
    * Fixed rule 3.1, 10.1, 8.6, 10.6, 10.3.
- 2.3.1:
  - Removed deprecated APIs.
- 2.3.0:
  - New feature:
    * Updated the driver code to adapt the case of interrupt configurations in GPIO module. New APIs were added to configure the GPIO interrupt settings if the module has this feature on it.
- 2.2.1:
  - 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 2.0.7.

- 2.0.7
  - Fixed the issue for MISRA-2012 check.
    * Fixed rule 11.9 ,15.7 ,14.4 ,10.4 ,10.8 ,10.3, 10.1, 10.6, 13.5, 11.3, 13.2, 17.7, 5.7, 8.3, 8.5, 11.1, 16.1.
  - Improvements:
    * Added I2C_MASTER_FACK_CONTROL macro to enable FACK control for master transfer receive flow when IP support double buffer, then master holds the SCL by manually setting TX AK/NAK during data transfer.

  - Fixed coverity issue of unchecked return value in I2C_RTOS_Transfer.
  - Bug fix:
    * Fixed variable redefine issue by moving i2cBases from fsl_i2c.h to fsl_i2c.c.
- 2.0.6
  - Bug fix:
    * Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) does not support the situation that master transfer with subaddress and transfer data size zero, which means no data follows by the subaddress.
- 2.0.5
  - 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 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 2.0.2.

- 2.0.2
  - Optimization
    * correct driver function LLWU_SetResetPinMode parameter name.
  - Bug Fix:
    * Fix MISRA-2012 rules.
      · Rule 14.4, 10.8, 10.4, 10.3.
- 2.0.1
  - Miscellaneous changes:
    * Updates for KL8x.
- 2.0.0
  - Initial version.

LPTMR

The current LPTMR driver version is 2.0.2.

- 2.0.2
  - Bug Fix:
    * Fix MISRA-2012 issues.
      · Rule 10.1.
- 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 2.2.7.

- 2.2.7
  - Fix the issue for MISRA-2012 check.
    * Fixed rule-12.1, rule-17.7, rule-14.4, rule-13.3, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3.
- 2.2.6
  - Fix the repeatedly reading status register issue while dealing with the IRQ routine.
- 2.2.5
  - Do not set or clear the TIE/RIE bits when using LPUART_EnableTxDMA() and LPUART_-

EnableRxDMA().
- 2.2.4
  - Added hardware flow control function support.
  - Added idle line detected feature in LPUART_TransferNonBlocking function. If an idle line was detected, a callback is triggered with status kStatus_LPUART_IdleLineDetected returned. This feature may be useful when the received Bytes is less than the expected receive data size. Before triggering the callback, data in the FIFO (if has FIFO) is read out, and all interrupts will not be disabled, except if the receive data size reaches 0.
  - Enabled the RX FIFO watermark function. With the idle line detected feature enabled, you can set the watermark value to whatever you want (should be less than the RX FIFO size). Data is received and a callback is triggered when data receive is end.
- 2.2.3
  - Changed parameter type in LPUART_RTOS_Init() struct rtos_lpuart_config –> lpuart_rtos_config_t.
  - Bug fix:
    - Disabled LPUART receive interrupt instead of disabling all NVIC when read data from ring buffer. Because the ring buffer is used, receive nonblocking disables all NVIC interrupts to protect the ring buffer. This has a negative effect to other IPS which are using the interrupt.
- 2.2.2
  - Added software reset feature support.
  - Added software reset API to LPUART_Init().
- 2.2.1
  - Added separate RX/TX IRQ number support.
- 2.2.0
  - Added 7 data bits and MSB support.
- 2.1.1
  - Removed needless check of event flags and assert in LPUART_RTOS_Receive.
  - Always wait for RX event flag in LPUART_RTOS_Receive.
- 2.1.0
  - Update transactional APIs.

PDB

The current PDB driver version is 2.0.2.

- 2.0.2
  - Improvement:
    - Use macros in feature file instead of that in the iomap.
- 2.0.1
  - Changed PDB register base array to const.
- 2.0.0
  - Initial version.

PIT

The current PIT driver version is 2.0.2.

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

PMC

The current PMC driver version is 2.0.1.

- 2.0.1 -Fix MISRA issu -Rule 10.8, rule 10.3.
- 2.0.0
  - Initial version.

PORT

The current PORT driver version is 2.1.0.

- 2.1.0
  - New feature:
    - Update the driver code to adapt the case of the interrupt configurations in GPIO module. will move the pin configuration APIs to GPIO module.
- 2.0.2
  - 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 2.0.2.

- 2.0.2
  - Fix MISRA issue -Rule 10.8, rule 10.1, rule 13.2, rule 3.1.
- 2.0.1
  - [KPSDK-10249] Fixed kRCM_SourceSw bit shift issue.
- 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 2.0.5.

- 2.0.5
  - Fix the issue for MISRA-2012 check.
    - Fixed rule 15.7, rule 14.4, rule 10.3, rule 10.1, rule 10.4.
- 2.0.4
  - When entering stop modes, use ram function for the flash synchronize issue. Application should make sure that, the rw data of fsl_smc.c is located in memory region which is not powered off in stop modes.
- 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.

UART

The current UART driver version is 2.1.6.

- 2.1.6
  - Fix the repeatly reading status register issue while dealing the IRQ routine.
- 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.

VREF

The current VREF driver version is 2.1.1.

- 2.1.1
  - MISRA-2012 issue fixed.
    - Fixed rule contain: rule-10.4, rule-10.3, rule-10.1.
- 2.1.0
  - Added new functions:
    - Supported L5K board: add VREF_SetTrim2V1Val() and VREF_GetTrim2V1Val() func-

tions to supply 2V1 output mode.
- 2.0.0
  - Initial version.

WDOG

The current WDOG driver version is 2.0.0.

- 2.0.0
  - Initial version.

CLOCK

Current CLOCK driver version is 2.3.0

- 2.3.0
  - Fix the issue for MISRA-2012 check.
    - Fixed rule 10.4, rule 10.1, rule 10.6, rule 13.5, rule 10.8.
  - Bug Fix:
    - Fix incorrect External Oscillator Configuration sequence and ensure oscillator configuration be executed before it be enabled.
  - New feature
    - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.2.1
  - Bug Fix:
    - Fix the issue that 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 Fix:
    - Update IP_CLOCKS array, remove 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 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 (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 fix:
    - 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

Current version is FatFs R0.13b_rev0.

- R0.13b_rev0
  - Upgraded to version 0.13b
- R0.13a_rev0

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

SDMMC

The current driver version is 2.2.11.

- 2.2.11
  - BugFix
    - Fix NULL pointer dereference issue when calling function SDMMCHOST_CardDetectInit in host adaptor layer.
    - Fix logical dead code issue in SDMMC_SwitchToVoltage function.
- 2.2.10
  - BugFix:
    - Add NUll pointer check for USDHC freertos adaptor transfer complete callback.
    - Add event value check for all the freertos event to fix program hang when a card event occur before create.
- 2.2.9
  - Improvement:
    - Add NULL pointer check for sdmmchostcard_usr_param_t member cd in card detect callback to avoid memory corruption.
    - Add card voltage switch function in sdmmhostcard_usr_param_t to allow application reigster card signal line voltage switch function.
  - Bug fix
    - Fix host freertos adaptor and polling adaptor can't detect card insert bug for usdhc.
    - Fix sdhc host layer build issue and typo issue.
- 2.2.8
  - Improvement:
    - Update sdmmc to support sdio interrupt.
- 2.2.7
  - BugFix:
    - Fix MDK 66-D warning.

- 2.2.6
  - Improvement:
    - Remove some soc specific header files from porting layer.
    - Save MMC OCR registers while sending CMD1 with argument 0.
  - Bugfix:
    - Add MMC_PowerOn function in which there is delay function after powerup sdcard.-otherwise,card may init failed.
- 2.2.5
  - New features:
    - Add SD_ReadStatus api to get 512bit SD status.
    - Add error log support in sdcard functions.
    - Add SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning and it is disabled by default.
    - Add error procedure in the transfer function to improve stability.
    - Remove deprecated gpio api in host layer.
- 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 response 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.
- 2.1.4
  - Miscellaneous:
    - Added Host reset function for card re-initialization.
    - 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 incorrect comparison between count and length in MMC_ReadBlocks/MMC_WriteBlocks.
- 2.1.1
  - Bug fix:
    - 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 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.

SDIO

The current driver version is 2.2.11.

- 2.2.11
  - Buf fix:
    - Add check card async interrupt capability in function SDIO_GetCardCapability.
    - Fix OUT OF BOUNDS access in function SDIO_IO_Transfer.
- 2.2.10
  - Bug fix:
    - Fix sdio card driver get wrong io number when the card io number is bigger than 2.
  - New feature:
    - Add sdio 3.0 support.
    - Add api SDIO_IO_RW_Direct for direct read/write card register access.
- 2.2.9
  - Improvement:
    - Add api SDIO_SetIOIRQHandler/SDIO_HandlePendingIOInterrupt to handle multi io pending IRQ.
- 2.2.8
  - Improvement:
    - Update sdmmc to support sdio interrupt.
    - Add api SDIO_GetPendingInterrupt to get the pending io interrupt.
- 2.2.7
  - Bug fix:
    - Fix MDK 66-D warning.
- 2.2.6
  - New features:
    - Add a unify transfer interface for SDIO.
  - Bug fix:
    - Wrong pointer address used by SDMMCHOST_Init.
- 2.1.5
  - Bug fix:
    - Improved SDIO card init sequence and add retry option for SDIO_SwitchToHighSpeed function.
- 2.1.4
  - Miscellaneous:
    - Added Go_Idle function for SDIO card.

- 2.0.0
  - Initial version.

SDSPI

The current driver version is 2.1.4.

- 2.1.4
  - Bug fix:
    - Fix MDK 66-D warning.
- 2.1.3
  - Improve sdspi code size and performance.
- 2.0.0
  - Initial version.

3 RTOS Change Log

FreeRTOS for MCUXpresso SDK.

The current version is Amazon-FreeRTOS 1.4.0 Original package is available at github.-com/aws/amazon-freertos.

- 1.4.7_rev0
  - New features:
    - Add optional allocation scheme heap_useNewlib.c by D. Nadler.
    - Enable task aware debugging for cm33 platforms
    - Move tickless implementation to application layer
  - Other changes:
    - Fix other build warnings, errors
- 1.4.6_rev0
  - New features:
    - Update support of CM33 port with Trustzone, MPU, FPU support
    - Add support for AWS test for Cypress WiFi
    - Use lwip netif api to avoid lwIP raw API calls outside of tcpip thread in aws_wifi.c
  - Other changes:
    - Fix issues with mflash driver
    - Fix other build warnings, errors
- 1.4.0_rev1
  - New features:
    - Add implementation of vTaskEndScheduler for CM0 GCC port.
    - Support for CM33, CM33F architectures based on CM3, CM4F ports
- 1.4.0_rev0
  - New features:
    - Support for pkcs11 for several platforms, secure element host library under pkcs11/portable/nxp folder
    - Lwip, wifi_qca support for secure_sockets in secure_sockets/portable/nxp folder
    - Flash driver support for several platforms in third_party/mcu_vendor/nxp folder
    - Generic support for aws_wifi under wifi/portable/nxp/common folder
  - Other changes:
    - Fix several build warnings, errors

Updates applied to FreeRTOS kernel up to version 10.0.0 (up to Amazon - FreeRTOS merge). New kernel related changes will be described in section above as part of AWS package.

- 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/Source/portable/low_power_tickless/.
- 9.0.0_rev2
  - New features:
    * Enabled MCUXpresso thread aware debugging. Add freertos_tasks_c_additions.h and configINCLUDE_FREERTOS_TASK_C_ADDITIONS_H and configFRTOS_MEMORY_SCHEME macros.
- 9.0.0_rev1
  - New features:
    * Enabled -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 the preferred solution. Not same as _weak!
- 8.2.3
  - New features:
    * Tickless idle mode support.
    * Added template application for Kinetis Expert (KEx) tool (template_application).
  - Other changes:
    * Folder structure reduction. Keep only Kinetis related parts.

How To Reach Us

Home Page:

nxp.com

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.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 06/2019
Document identifier: MCUXSDKKV31RN

arm