Document Number: MCUXSDKIMX8QXRN
Rev. 0, 02/2019

MCUXpresso SDK Release Notes for i.MX 8QuadXPlus

Contents

1 Overview

The MCUXpresso Software Development Kit (MCUXpresso SDK) is a collection of software enablement for Microcontrollers (MCU) that includes peripheral drivers, other middleware packages, such as multicore support 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.

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpresso SDK is the evolution of Kinetis SDK v2.0.0, which 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.

NOTE

In order to maintain compatibility with legacy Freescale code, the filenames and source code in MCUXpresso SDK containing the legacy

Review Draft
Confidential Proprietary

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 8.11.2
- Makefiles support with GCC revision v5-2016-q3 from Arm Embedded
- System Controller Firmware (SCFW): Version 0.2 from the NXP website

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 devices and development boards

Development boards	devices
i.MX 8QXP MEK CPU board, i.MX 8QMax, i.MX8QXP MEK base board	**PIMX8QX6AVLFZAA, PIMX8QX6AVLFZAB, PIMX8QX2AVOFZAB**

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
Driver examples	<install_dir>/boards/<board_name>/driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
CMSIS Driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
mmCAU examples	<install_dir>/boards/<board_name>/mmcau_examples
Multicore examples	<install_dir>/boards<board_name>/multicore_examples
Documentation	<install_dir>/docs
lWIP Documentation	<install_dir>/docs/lwip
Middleware	<install_dir>/middleware
mmCAU	<install_dir>/middleware/mmcau_<version>
Multicore stack	<install_dir>/middleware/multicore_<version>

Table continues on the next page...

Table 2. Release contents (continued)

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, devices, documentation, and RTOS support.

6.1 Device support

The device folder contains available software enablement for the specific SoC subfamily. This folder includes clock-specific implementation, device register header file, device register feature header file, 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 System-on-Chip (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, and RTOS 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 folder contains examples demonstrating the use of the included source.

NOTE

Some demo applications and driver examples are intended for a single ARM Cortex-M4 application reference. They cannot support running with the Linux BSP, which requires additional service protocol implementation. See the readme file for the specific application to know whether it supports running with the Linux BSP.

6.2 Middleware

6.2.1 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.2 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 C 2012 rules with the following exceptions.

Table 3. 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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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.
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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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

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 Break point cannot work when debugging application located in DDR with System Bus cache enabled for A0 silicon

This is debugger's restriction. The DDR is connected to the system bus for Cortex-M4, but the hardware BPT is not available on the system bus, so the software BPT cannot work with cache enabled. Disabling the system bus cache can make the debugger work. To disable the system bus cache, comment out the following lines in the "SystemInit(void)" function located in *devices/<device_name>/system_<device_name>_cm4_<core_index>.c*.

```
LMEM->PSCCR |= (LMEM_PSCCR_ENWRBUF_MASK | LMEM_PSCCR_ENCACHE_MASK);
```

MCUXpresso SDK Release Notes for i.MX 8QuadXPlus

Change Logs

Contents

Contents

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.1.0

- 2.1.0
  - New Features:
    - Update clock_ip_name_t, adding LPCG base information.
    - Add LPCG configure API CLOCK_ConfigLPCG/CLOCK_SetLpcgGate.
    - Update CLOCK_EnableClockExt/CLOCK_DisableClock API, adding LPCG gate control.
- 2.0.1
  - some minor fixes.
- 2.0.0
  - initial version.

DPU

Current DPU driver version is 2.0.0

- 2.0.0
  - Initial version.

DPU IRQSTEER

The current DPU IRQSTEER driver version is 2.0.0.

- 2.0.0
  - Initial version.

EDMA

The current EDMA driver version is 2.1.0.

- 2.1.0
  - Bugfix
    - Add const type for parameter config in EDMA_SubmitTransfer and EDMA_HandleTransferConfig API.
    - Add configurations for srcAddr and destAddr in EDMA_PrepareTransfer API.
- 2.0.2
  - Update EDMA driver to support MP_CR bit GMRC.
  - Update EDMA instance name for i.MX 8QM.

  - Use instance number as factor to calculate channel number for different instance instead of hard code.
- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
  - Add s_EDMAEnabledChannel to record enabled channel to merge all the channel IRQ handler into driver IRQ handler.
  - Remove all the separated channel IRQ handler in DMA driver.
  - Add feature macro for bits EMI and EBW in MP_CSR.
- 2.0.0
  - Initial version.

ENET

The current ENET driver version is 2.2.3.

- 2.2.3
  - Improved data buffer cache maintenance in the ENET driver.
- 2.2.2
  - Added the APIs for extended multi-ring support.
  - Added the AVB configure API for extended AVB feature support.
- 2.2.1
  - Changed the input data pointer attribute to const in ENET_SendFrame().
- 2.1.1
  - Added the extended MDIO IEEE802.3 Clause 45 MDIO format SMI command APIs.
  - Added the extended interrupt coalescing feature.
  - Combined all storage operations in the ENET_Init to ENET_SetHandler API.
- 2.0.1
  - Bug fix:
    - Used direct transmit busy check when doing data transmit.
  - Miscellaneous changes:
    - Updated IRQ handler work flow.
    - Changed the TX/RX interrupt macro from kENET_RxByteInterrupt to kENET_RxBufferInterrupt, from kENET_TxByteInterrupt to kENET_TxBufferInterrupt.
    - Deleted unnecessary parameters in ENET handler.
- 2.0.0
  - Initial version.

ESAI

The current ESAI driver version is 2.0.1.

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

  - Initial version.

FLEXCAN

The current FLEXCAN driver version is 2.3.1.

- 2.3.1
  - MISRA C-2012 issue check.
    - Fixed rule contain: rule-12.1, rule-17.7, rule-16.4, rule-11.9, rule-8.4, rule-14.4, rule-10.8, rule-10.4, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-8.3, rule-12.2. rule-16.1.
  - Improvements:
    - Implementation for ERR009595.
    - Implementation for ERR005829.
    - Implementation for ERR006032.
    - fix issue for [FlexCAN] RemoteRequest UT Case run fail
    - Implementation all TX and RX transfering Timestamp used in FlexCAN demos
    - fix issue for UT Test Fail for CANFD payload size changed from 64BperMB to 8PerMB
    - Implementation for Update released CAN-FD capable products to same driver support, improve baud rate API
  - Bug fixes:
    - Adding correct handle when kStatus_FLEXCAN_TxSwitchToRx comming.
    - CANFD transfer data fail when use bus baudrate 30Khz
    - fix issue ERR009595 not folllow the ERRATA document.
    - fix code error for ERR006032 work around solution
- 2.3.0
  - Improvements:
    - Added self-wakeup support from STOP mode in the interrupt handling.
- 2.2.3
  - Bug fix:
    - Fixed CANFD data phase's bit rate not set as expected.
- 2.2.2
  - Improvements:
    - Added a time stamp feature and enabled it in the interrupt_transfer example.
- 2.2.1
  - Improvements:
    - Separated CANFD initialization API.
    - In the interrupt handling, fixed the issue that the user cannot use the normal CAN API when FD is present.
- 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 incorrect function name spelling of FLEXCAN_XXX()
    * Moved Freeze Enable/Disable setting from FLEXCAN_Enter/ExitFreezeMode() to FLEXCAN_Init();
    * Fixed incorrect helper macro values.
  - Other changes:
    * Hid FLEXCAN_Reset() from user.
    * Used NDEBUG macro to wrap FLEXCAN_IsMbOccupied() function instead of DEBUG macro.
- 2.0.0
  - Initial version.

FLEXSPI

The current FLEXSPI driver version is 2.0.5.

- 2.0.5
  - Bug fixes:
    * Fix FLEXSPI_UpdateLUT cannot do partial update issue .
- 2.0.4
  - Bug fixes:
    * Reset flash size for all ports to zero in FLEXSPI_Init, fixed the possible out of range flash access with no error reported.
- 2.0.3
  - Bug fixes:
    * Fixed AHB receive buffer size configuration issue. The FLEXSPI_AHBRXBUFCR0_BUFSZ field should configure 64 bits size, and currently the the AHB receive buffer size is in bytes which means 8-bit, so the correct configuration should be config->ahbConfig.buffer[i].bufferSize / 8.
- 2.0.2
  - New features:
    * Supports DQS write mask enable/disable feature during set FLEXSPI configuration.
    * Provides new API FLEXSPI_TransferUpdateSizeEDMA for user to update eDMA transfer size(SSIZE/DSIZE) per DMA transfer.
  - Bug fixes:
    * Fixed FLEXSPI_Init invalid operation to enable AHB bus Read Access to IP RX FIFO issue.
    * Fixed FLEXSPI_Init incorrect operation to configure IP TX FIFO watermark issue.
- 2.0.1
  - Bug fixes:
    * Fixed the flag clear issue and AHB read Command index configuration issue in FLEXSPI_SetFlashConfig.
    * Updated FLEXSPI_UpdateLUT function to update LUT table from any index instead of

previous command index.
    - Added bus idle wait in FLEXSPI_SetFlashConfig and FLEXSPI_UpdateLUT to ensure bus is idle before any change to FlexSPI controller.
    - Updated interrupt API FLEXSPI_TransferNonBlocking and interrupt handle flow FLEXSPI_TransferHandleIRQ.
    - Updated eDMA API FLEXSPI_TransferEDMA.
- 2.0.0
  - Initial version.

FTM

The current FTM driver version is 2.1.0.

- 2.1.0
  - New feature:
    - Add 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 will not set the timer period(MOD value), it will be useful for users to set the PWM parameters without changing the timer period.
  - Bug fixes:
    - Add 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.

GPT

The current GPT driver version is 2.0.0.

- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.0.1.

- 2.0.1:
  - API interface changes:
    - Refined naming of API while keeping all original APIs, marking them as deprecated. Original API will be removed in next release. The main change is to update the API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

INTMUX

The current INTMUX driver version is 2.0.1.

- 2.0.1
  - Improvements:
    - Added weak function implementations of INTMUX1_x_DriverIRQHandler, x ranges from 0 to 7 for supporting 8 channels.
- 2.0.0
  - Initial version.

IRQSTEER

The current IRQSTEER driver version is 2.0.1.

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

ISI

The current ISI driver version is 2.0.1.

-2.0.1

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

2.0.0

- Initial version.

LDB (LVDS Display Bridge)

The current LDB driver version is 2.0.0.

- 2.0.0
  - Initial version.

CACHE

The current CACHE driver version is 2.0.1.

- 2.0.1
  - Fixed the over 4 KB size maintenance issue in invalidate/clean/clean&invalidate by range APIs.
- 2.0.0
  - Initial version.

LPADC

The current LPADC driver version is 2.0.3.

- 2.0.3
  - Improvements:
    - Add the API LPADC_SetOffsetValue() to support configure offset trim value manually.
    - Add the API LPADC_DoOffsetCalibration() to do offset calibration independently.
    - Improve the usage of macros and remove invalid macros.
- 2.0.2
  - Add supports for platforms with 2 FIFOs and different calibration measures.
- 2.0.1
  - Ensure the API LPADC_SetConvCommandConfig configure related registers correctly.
- 2.0.0
  - Initial version.

LPI2C

The current LPI2C driver version is 2.1.6.

- 2.1.6
  - Bug fix:

    - Fix driver MISRA build error and c++ build error in LPI2C_MasterSend and LPI2C_-SlaveSend.
- 2.1.5
  - Bug fix:
    - Extended the Driver IRQ handler to support LPI2C4 and change to use ARRAY_SIZE(k-Lpi2cBases) instead of FEATURE COUNT to decide the array size for handle pointer array.
- 2.1.4
  - Bug fix:
    - Fixed the LPI2C_MasterTransferEDMA receive issue when LPI2C share same request source for TX/RX DMA request. Previously, the API uses scatter-gather method, which handles the command transfer first, then handles the linked TCD which is preset with the receive data transfer. The issue is that the TX DMA request and the RX DMA request are both enabled, so when the DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happens due to TX FIFO empty. The result is the RX DMA transfer starts, without waiting on the expected RX DMA request.
    - Fixed the issue by enabling IntMajor interrupt for the command TCD and checking if there is a linked TCD to disable the TX DMA request in LPI2C_MasterEDMACallback API.
- 2.1.3
  - Improvement:
    - Added LPI2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
    - Added LPI2C_MasterTransferBlocking API.
- 2.1.2
  - Bug fix:
    - In LPI2C_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag is detected.
- 2.1.1
  - Bug fix:
    - Disabled auto stop feature in eDMA driver. Previously, the auto-stop feature was enabled at transfer when transferring with stop flag. If the previous transfer was without stop flag, because the auto stop feature is enabled, then when starting a new transfer with stop flag, the stop flag sends before starting the new transfer, and the start flag cannot successfully send, so the transfer cannot start.
    - Changed default slave configuration with address stall false.
- 2.1.0
  - API name change:
    - LPI2C_MasterTransferCreateHandle -> LPI2C_MasterCreateHandle.
    - LPI2C_MasterTransferGetCount -> LPI2C_MasterGetTransferCount.
    - LPI2C_MasterTransferAbort -> LPI2C_MasterAbortTransfer.
    - LPI2C_MasterTransferHandleIRQ -> LPI2C_MasterHandleInterrupt.
    - LPI2C_SlaveTransferCreateHandle -> LPI2C_SlaveCreateHandle.
    - LPI2C_SlaveTransferGetCount -> LPI2C_SlaveGetTransferCount.
    - LPI2C_SlaveTransferAbort -> LPI2C_SlaveAbortTransfer.

        * LPI2C_SlaveTransferHandleIRQ -> LPI2C_SlaveHandleInterrupt.
- 2.0.0
  - Initial version.

LPIT

The current LPIT driver version is 2.0.0.

- 2.0.0
  - Initial version.

LPSPI

The current LPSPI driver version is 2.0.3.

- 2.0.3
  - Bug Fix:
    * Remove the LPSPI_Reset() from LPSPI_MasterInit() and LPSPI_SlaveInit(), because this API may glitch the slave select line, if needed, please call this function manually.
- 2.0.2
  - New feature:
    * Added dummy data setup API to allow users to configure the dummy data to be transferred.
    * Enabled the 3-wire mode, SIN and SOUT pins can be configured as input/output pin.
- 2.0.1
  - Bug fix:
    * The clock source should be divided by the PRESCALE setting in LPSPI_MasterSetDelayTimes function.
    * Fixed the bug that LPSPI_MasterTransferBlocking function would hang in some corner cases.
  - Optimization:
    * Added #ifndef/#endif to allow user to change the default TX value at compile time.
- 2.0.0
  - Initial version.

LPUART

The current LPUART driver version is 2.2.7.

- 2.2.7
  - Fix the issue for MISRA-2012 check.
    * Fixed rule-12.1, rule-17.7, rule-14.4, rule-13.3, rule-14.4, rule-10.4, rule-10.8, rule-10.3, rule-10.7, rule-10.1, rule-11.6, rule-13.5, rule-11.3, rule-13.2, rule-8.3.
- 2.2.6

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

CSI2RX

The current CSI2RX driver version is 2.0.0.

- 2.0.0
  - Initial version.

MIPI_DSI

The current MIPI_DSI driver version is 2.0.0.

- 2.0.0
  - Initial version.

MU

The Current MU driver version is 2.0.2.

- 2.0.2
  - Added support for MIMX8MQx.
- 2.0.1
  - Added support for MCIMX7Ux_M4.
- 2.0.0
  - Initial version.

RGPIO

The current RGPIO driver version is 2.0.1.

- 2.0.1
  - API Interface Change:
    - Refined naming of API while keep all original APIs with marking them as deprecated. The original API will be removed in next release. The main change is to update API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.2.0.

- 2.2.0
  - Improvement:
    - Add new apis for parameters collection and user interfaces simplify: SAI_Init SAI_SetMasterClockConfig SAI_TxSetBitClockRate SAI_TxSetSerialDataConfig SAI_TxSetFrameSyncConfig SAI_TxSetFifoConfig SAI_TxSetBitclockConfig SAI_TxSetConfig SAI_TxSetTransferConfig SAI_RxSetBitClockRate SAI_RxSetSerialDataConfig SAI_RxSetFrameSyncConfig SAI_RxSetFifoConfig SAI_RxSetBitclockConfig SAI_RXSetConfig SAI_RxSetTransferConfig SAI_GetClassicI2SConfig SAI_GetLeftJustifiedConfig SAI_GetRightJustifiedConfig S-

AI_GetTDMConfig

- 2.1.9
  - Improvement:
    - Improve sai driver comment for clock polarity.
    - Add enum for sai for sample inputs on different edge.
    - Change FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHANNEL_COUNTn(base) for the difference between the different sai instance.
    - Add new api SAI_TxSetBitClockDirection SAI_RxSetBitClockDirection SAI_RxSetFrameSyncDirection SAI_TxSetFrameSyncDirection
- 2.1.8
  - Improvement:
    - Add feature macro test for the sync mode2 and mode 3.
    - Add feature macro test for masterClockHz in sai_transfer_format_t.
- 2.1.7
  - Improvement:
    - Add feature macro test for the mclkSource member in sai_config_t.
    - Change "FSL_FEATURE_SAI5_SAI6_SHARE_IRQ" to "FSL_FEATURE_SAI_SAI5_SAI6_SHARE_IRQ".
    - Add #ifndef #endif check for SAI_XFER_QUEUE_SIZE to allow redefinition.
  - Bug fix:
    - Fix the build error caused by feature macro test for mclkSource.

-2.1.6

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

-2.1.5

- Bug fix:
  - Correct SAI3 driver IRQ handler name.
  - Add I2S4/5/6 IRQ handler.
  - Add base in handler structure to support different instances share one IRQ number.
- New feature:
  - Update SAI driver for MCR bit MICS.
  - Added 192KHZ/384KHZ in the sample rate enumeration.
  - Added multi FIFO interrupt/SDMA transfer support for TX/RX.
  - Added API to read/write multi FIFO data in a blocking method.
  - Added bclk bypass support when bclk is same with mclk.

2.1.4

- New feature:
  - Added API to enable/disable auto FIFO error recovery in platforms that support this feature.
  - Added API to set data packing feature in platform which support this feature.

2.1.3

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

2.1.2

- Bug fix:
  - Added 24-bit support for SAI eDMA transfer. All data shall be 32 bits for send/receive, as eDMA cannot directly handle 3 Byte transfer.

2.1.1

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

2.1.0

- API name change:
  - SAI_GetSendRemainingBytes -> SAI_GetSentCount.
  - SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount.
  - All transactional API name add "Transfer" prefix.
  - All transactional API use base and handle as input parameter.
  - Unify the parameter names.
- Bug fix:
  - Fixed WLC bug while reading TCSR/RCSR registers.
  - Fixed MOE enable flow issue, move MOE enable after MICS settings in SAI_TxInit/SAI_RxInit.

2.0.0

- Initial version.

SEMA42

The current SEMA42 driver version is 2.0.0.

- 2.0.0
  - Initial version.

TPM

The current TPM driver version is 2.0.3.

- 2.0.3
  - MISRA-2012 issue fixed.
    - Fixed rule contain: rule-12.1, rule-17.7, rule-16.3, rule-14.4 ,rule-1.3, rule-10.4, rule-10.3, rule-10.7, rule-10.1, rule-10.6, rule-18.1.
- 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 cannot set FILTER register issue.
    - Fix TPM_UpdateChnEdgeLevelSelect ACK wait issue.
- 2.0.0
  - Initial version.

TSTMR

The current TSTMR 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 its configuartion time period
      - Added WDOG32_Init API to quick access section.
      - Defined register variable in WDOG32_Init API.
- 2.0.0
  - Initial version.

2 Middleware Change Log

MMCAU library

The current version is 2.0.1.

- 2.0.1
  - Bug fixes:
    - KPSDK-17133 fix bug in fsl_mmcau.c when AES key schedule array is not aligned.
- 2.0.0
  - New features:
    - Q4/2013 release of the CAU library.
    - Added fsl_mmcau.h/fsl_mmcau.c optional layer between application and legacy CAU library (cau_api.h). This API has no alignment requirements.

lwIP for MCUXpresso SDK

The current version of lwIP is based on lwIP 2.1.2 and lwIP-contrib 2.1.0.

- 2.1.2_rev1
  - New features:
    - Ported lwIP 2.1.2 (2018-11-22, SHA-1: 159e31b689577dbf69cf0683bbaffbd71fa5ee10) to KSDK 2.0.0.
    - Ported lwIP-contrib 2.1.0 (2018-09-24, SHA-1: 35b011d4cf4c4b480f8859c456587a884ec9d287) to KSDK 2.0.0.
- 2.0.3_rev1
  - New features:
    - Ported lwIP 2.0.3 (2017-09-15, SHA-1: 92f23d6ca0971a32f2085b9480e738d34174417b) to KSDK 2.0.0.
- 2.0.2_rev1
  - New features:
    - Ported lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef) to KSDK 2.0.0.
- 2.0.0_rev3
  - New features:
    - Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0.
- 2.0.0_rev2
  - New features:
    - Ported lwIP 2.0.0 RC2 (2016-08-08, SHA-1: b1dfd00f9233d124514a36a8c8606990016f2ad4) to KSDK 2.0.0.
- 2.0.0_rev1
  - New features:
    - Ported lwIP 2.0.0 RC0 (2016-05-26) to KSDK 2.0.0.

        - Changed lwIP bare-metal examples to use poll-driven approach instead of interrupt-driven one.
- 1.4.1_rev2
  - New features:
    - Enabled critical sections in lwIP.
  - Bug fixes:
    - Fixed default lwIP packet-buffer size to be able to accept a maximum size frame from the ENET driver.
    - Fixed possible drop of multi-frame packets during transmission.
- 1.4.1_rev1
  - New features:
    - Ported lwIP 1.4.1 to KSDK 2.0.0.

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.

- 1.4.0_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_MEMO-RY_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.

© 2019 NXP B.V.

Document Number MCUXSDKIMX8QXRN
Revision 0, 02/2019