NXP Semiconductors
Release Notes

Document identifier: MCUXSDKMIMXRT1011RN
Rev. 0, 10/2019

MCUXpresso SDK Release Notes Supporting MIMXRT1010-EVK

Contents

Chapter 1 Overview.. 3

Chapter 2 MCUXpresso SDK.. 4

Chapter 3 Development tools.. 5

Chapter 4 Supported development systems..6

Chapter 5 Release contents...7

Chapter 6 MCUXpresso SDK release package...8

Chapter 7 MISRA compliance.. 11

Chapter 8 Known issues.. 14

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 USB and other middleware packages, such as multicore support and FatFs, and integrated RTOS support for FreeRTOS^{TM} 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 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 easy cloning of existing SDK examples and demos, allowing users to 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 8.40.2
- MDK-Arm Microcontroller Development Kit (Keil)® 5.27
- Makefiles support with GCC revision 8-2018-q4-major from Arm Embedded
- MCUXpresso IDE v11.0.1

Chapter 4
Supported development systems

This release supports boards and devices listed in the following table. The boards and devices in bold were tested in this release:

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
MIMXRT1010-EVK	MIMXRT1011CAE4A, **MIMXRT1011DAE5A**

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
Qualcomm WiFi	<install_dir>/middleware/wifi_qca
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
Cortex Microcontroller Software Interface Standard (CMSIS) driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Qualcomm WiFi stack examples	<install_dir>/boards/<board_name>/wifi_qca_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
USB stack	<install_dir>/middleware/usb
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

Chapter 6
MCUXpresso SDK release package

The MCUXpresso SDK release package content is 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 the whole software enablement available for the specific System-on-Chip (SoC) subfamily. This folder includes clock-specific implementation, device register header files, device register feature header files, 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 standard debug console.

The device-specific header files provide a direct access to the microcontroller peripheral registers. The device header file provides an overall SoC memory mapped register definition. The folder also includes the feature header file for each peripheral on the microcontroller.

The toolchain folder contains the startup code and linker files for each supported toolchain. The startup code is a CMSIS compliant startup code 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 USB stack

See the *MCUXpresso SDK USB Stack User's Guide* (document MCUXSDKUSBSUG) for more information.

6.2.1.1 Peripheral devices tested with USB Host stack

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

Table 3. Peripheral devices

Device type	Device

Table continues on the next page...

Table 3. Peripheral devices (continued)

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 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

Table continues on the next page...

Table 3. Peripheral devices (continued)

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 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 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.
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.

Table continues on the next page...

Table 4. MISRA exceptions (continued)

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 17.1	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.

Chapter 8 Known issues

8.1 Maximum file path length in Windows 7® operating system

The 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 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).

8.3 New Project Wizard compile failure

The following components request the user to manually select other components that they depend upon in order to compile. These components depend on several other 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 MIMXRT1010-EVK

Change Logs

Contents

Driver Change Log 1

CLOCK 1

IOMUXC 1

LPI2C_CMSIS 1

LPSPI_CMSIS 2

LPUART_CMSIS 2

ADC 2

ADC_ETC 3

AIPSTZ 3

AOI 3

CACHE 3

COMMON 3

DCDC 4

DMAMUX 4

EDMA 5

EWM 7

FLEXIO 7

FLEXIO_UART 7

FLEXIO_I2C 8

FLEXIO_SPI 9

FLEXIO_I2S 10

Contents

Title | Page Number

FLEXIO_MCU_LCD 11

FLEXIO_CAMERA 11

FLEXRAM 12

FLEXSPI 12

GPC 13

GPT 14

GPIO 14

KPP 14

LPI2C 15

LPSPI 17

LPUART 17

PIT 18

PMU 19

PWM 19

RTWDOG 19

SAI 20

SPDIF 22

SRC 22

TEMPMON 22

WDOG 22

XBARA 23

Middleware Change Log 24

FatFs for MCUXpresso SDK 24

USB stack for MCUXpresso SDK 24

QCA WiFi 28

Contents

Title | Page Number

RTOS Change Log 29

FreeRTOS for MCUXpresso SDK. 29

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.3.0.

- 2.3.0
  - New feature:
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.2.0
  - New feature
  - Adding new API CLOCK_DelayAtLeastUs() implemented by DWT to allow users set delay in unit of microsecond.
- 2.1.6
  - Bug Fix:
    - Fix build issue with GCC compiler when include header from C++ file.
- 2.1.5
  - Bug Fix:
    - Add initialization of the fractional mode and spread spectrum mode in CLOCK_InitSysPll().
- 2.1.4
  - Optimization:
    - Add PerClk in clock_name_t and CLOCK_GetFreq.
    - Add APIs to get the frequency of AHB clock and SEMC, IPG clock and PER clock.
- 2.1.3
  - Use double instead of uint64_t to achieve better performance with double precision FPU.
- 2.1.2
  - some minor fixes.
- 2.0.0
  - initial version.

IOMUXC

The current IOMUXC driver version is 2.0.0.

- 2.0.0
  - initial version.
- 2.0.1
  - Delete enum value kIOMUXC_GPR_USBExposureMode in the _iomuxc_gpr_mode.

LPI2C_CMSIS

Current LPI2C_CMSIS driver version is 2.0

- 2.0
  - Initial version.

LPSPI_CMSIS

Current LPSPI_CMSIS driver version is 2.1

- 2.1
  - Bug Fix:
    - Fix the incorrect clock polarity assignment in the driver. For ARM_SPI_CPOL0_CPHA0 and other frame format parameters, CPOL = 0 means kSPI_ClockPolarityActiveHigh not kSPI_ClockPolarityActiveLow in driver.
  - New feature:
    - Allow user to set up the default Transmit value by using ARM_SPI_SET_DEFAULT_TX_VALUE. Please note that set default value is not supported in slave interrupts, because the pin will stay tristated if TX buffer is NULL.
    - Enable slave select mode in the new driver, but this has no effect when users set any of them because the driver can only support the hardware control function.
    - Enable 3-Wire mode, users can use ARM_SPI_MODE_MASTER_SIMPLEX/ARM_SPI_MODE_SLAVE_SIMPLEX to enable this feature. For ARM_SPI_MODE_MASTER_SIMPLEX mode, select SOUT pin as the input/output pin, and for ARM_SPI_MODE_SLAVE_SIMPLEX, the SIN pin is selected as the input/output pin.
- 2.0
  - - Initial version.

LPUART_CMSIS

Current LPUART_CMSIS driver version is 2.0

- 2.0
  - Initial version.

ADC

The current ADC driver version is 2.0.2.

- 2.0.2
  - Improvements
    - Used conversion control feature macro instead of that in IO map.
- 2.0.1
  - New Features
    - Added a control macro to enable/disable CLOCK code in current driver.
- 2.0.0
  - Initial version.

ADC_ETC

The current ADC_ETC driver version is 2.1.0.

- 2.1.0
  - New Features
    - Supported independent IRQ enable bit in ADC-ETC chain configuration registers.
    - Supported trigger n DONE3 interrupt operations.
- 2.0.1
  - New Features
    - Added a control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

AIPSTZ

The current AIPSTZ driver version is 2.0.0.

- 2.0.0
  - Initial version.

AOI

The current AOI driver version is 2.0.0.

- 2.0.0
  - Initial version.

CACHE

The current CACHE driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed cache size issue in L2CACHE_GetDefaultConfig API.
- 2.0.0
  - Initial version.

COMMON

The current COMMON driver version is 2.2.0.

- 2.2.0
  - New Features
    - Moved SDK_DelayAtLeastUs function from clock driver to common driver.

- 2.1.4
  - New Features
    - Added OTFAD into status group.
- 2.1.3
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.3.
- 2.1.2
  - Improvements
    - Add SUPPRESS_FALL_THROUGH_WARNING() macro for the usage of suppressing fallthrough warning.
- 2.1.1
  - Bug Fixes
    - Deleted and optimized repeated macro.
- 2.1.0
  - New Features
    - Added IRQ operation for XCC toolchain.
    - Added group IDs for newly supported drivers.
- 2.0.2
  - Bug Fixes
    - MISRA C-2012 issue fixed.
      - Fixed the rule: rule-10.4.
- 2.0.1
  - Improvements
    - Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
    - Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_S-ECTION" for specific SoCs which have no noncacheable sections, that helps avoid an unnecessary complex in link file and the startup file.
    - Updated the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
  - Initial version.

DCDC

The current DCDC driver version is 2.0.0.

- 2.0.0
  - Initial version.

DMAMUX

The current DMAMUX driver version is 2.0.3.

- 2.0.3

  - Bug Fixes
    - Fixed the issue for MISRA-2012 check.
      - Fixed rule 10.4 and rule 10.3.
- 2.0.2
  - New Features
    - Added an always-on enable feature to a DMA channel for ULP1 DMAMUX support.
- 2.0.1
  - Bug Fixes
    - Fixed the build warning issue by changing the type of parameter source from uint8_t to uint32_t when setting DMA request source in DMAMUX_SetSourceChange.
- 2.0.0
  - Initial version.

EDMA

The current eDMA driver version is 2.2.0.

- 2.2.0
  - Improvements
    - Added peripheral-to-peripheral support in EDMA driver.
- 2.1.9
  - Bug Fixes
    - Fixed MISRA issue: Rule 10.7 and 10.8 in function EDMA_DisableChannelInterrupts and EDMA_SubmitTransfer.
    - Fixed MISRA issue: Rule 10.7 in function EDMA_EnableAsyncRequest.
- 2.1.8
  - Bug Fixes
    - Fixed incorrect channel preemption base address used in EDMA_SetChannelPreemption-Config API which causes incorrect configuration of the channel preemption register.
- 2.1.7
  - Bug Fixes
    - Fixed incorrect transfer size setting.
      - Added 8 bytes transfer configuration and feature for RT series;
      - Added feature to support 16 bytes transfer for Kinetis.
    - Fixed the issue that EDMA_HandleIRQ would go to incorrect branch when TCD was not used and callback function not registered.
- 2.1.6
  - Bug Fixes
    - Fixed KW3X MISRA Issue.
      - Rule 14.4, 10.8, 10.4, 10.7, 10.1, 10.3, 13.5, and 13.2.
  - Improvements
    - Cleared the IRQ handler unavailable for specific platform with macro FSL_FEATURE_-EDMA_MODULE_CHANNEL_IRQ_ENTRY_SHARED_OFFSET.
- 2.1.5

- Improvements
  - Improved EDMA IRQ handler to support half interrupt feature.
- 2.1.4
  - Bug Fixes
    - Cleared enabled request, status during EDMA_Init for the case that EDMA is halted before reinitialization.
- 2.1.3
  - Bug Fixes
    - Added clear DONE bit in IRQ handler to avoid overwrite TCD issue.
    - Optimized 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_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 Fixes
    - Fixed the incorrect pubweak IRQHandler name issue, which caused re-definition build errors when client set his/her own IRQHandler, by changing the 32-channel IRQHandler name to DriverIRQHandler.
- 2.0.2
  - Bug Fixes
    - Fixed incorrect minorLoopBytes type definition in _edma_transfer_config struct, and defined minorLoopBytes as uint32_t instead of uint16_t.
- 2.0.1
  - Bug Fixes
    - 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
  - Bug Fixes
    * Fixed the hard fault in EWM_Deinit.
- 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:
    * Fixed 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 2.1.5.

- 2.1.5
  - Trigger user callback after all the data in ringbuffer are received in FLEXIO_UART_TransferReceiveNonBlocking.
- 2.1.4
  - Unified 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 configuration instead of disable module and disable clock.
    * Updated module Enable APIs to only support enable operation.
- 2.1.2
  - Bug fixes:
    * Fixed the transfer count calculation issue in FLEXIO_UART_TransferGetReceiveCount,

        FLEXIO_UART_TransferGetSendCount, FLEXIO_UART_TransferGetReceiveCountDMA, FLEXIO_UART_TransferGetSendCountDMA, FLEXIO_UART_TransferGetReceiveCountEDMA and FLEXIO_UART_TransferGetSendCountEDMA
    - Fixed the Dozen mode configuration error in FLEXIO_UART_Init API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.
    - Reported error when set baudrate too low and FLEXIO cannot reach that baudrate.
    - Disabled FLEXIO_UART receive interrupt instead of disable all NVIC when read data from ring buffer. Because ring buffer is used, receive nonblocking disables all NVIC interrupts to protect the ring buffer. This has negative effects on other IPS which are using interrupt.
- 2.1.1
  - Bug fixes:
    - Changed the API name FLEXIO_UART_StopRingBuffer to FLEXIO_UART_TransferStopRingBuffer to align with the definition in C file.
- 2.1.0
  - New features:
    - Added Transfer prefix in transactional APIs.
    - Added txSize/rxSize in handle structure to record the transfer size.
  - Bug fixes:
    - Added error handle to handle the data count is zero or data buffer is NULL situation.

FLEXIO_I2C

The current FLEXIO_I2C driver version is 2.1.8.

- 2.1.8
  - Fixed Coverity issue of useless call in FLEXIO_I2C_MasterTransferRunStateMachine.
- 2.1.7
  - New feature:
    - Added API of checking bus pin status.
  - Bug fixes:
    - Fixed the issue that FLEXIO_I2C_MasterTransferBlocking does not wait for STOP bit sent.
    - Fixed COVERITY issue of useless call in FLEXIO_I2C_MasterTransferRunStateMachine.
    - Fixed the issue that I2C master does not check whether bus is busy before transfer.
- 2.1.6
  - Bug fix:
    - Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) do not support the situation of master transfer with subaddress and transfer data size zero, which means no data follows the subaddress.
- 2.1.5
  - Unified 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 disabling module/clock.
    - Updated module Enable APIs to only support enable operation.
- 2.1.3
  - Changed the prototype of FLEXIO_I2C_MasterInit to return kStatus_Success if initialization successfully and return kStatus_InvalidArgument if "(srcClock_Hz / masterConfig->baudRate_Bps) / 2 - 1" exceeds 0xFFU.
- 2.1.2
  - Fixed the FLEXIO I2C issue where the master cannot receive data from I2C slave in high baudrate.
  - Fixed the FLEXIO I2C issue where the master cannot receive NAK when master sends non-existent addr.
  - Fixed the FLEXIO I2C issue where the master cannot get transfer count successfully.
  - Fixed the FLEXIO I2C issue where the master cannot receive data successfully when sending data first.
  - Fixed the Dozen mode configuration error in FLEXIO_I2C_MasterInit API. For enableInDoze = true, the configuration should be 0; for enableInDoze = false, the configuration should be 1.
  - Fixed the FLEXIO_I2C_MasterTransferBlocking API calls FLEXIO_I2C_MasterTransferCreateHandle issue. This leads the s_flexioHandle/s_flexioIsr/s_flexioType variable written. Then, if calling FLEXIO_I2C_MasterTransferBlocking API multiple times, the s_flexioHandle/s_flexioIsr/s_flexioType variable cannot be written any more 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
  - Unified 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 disabling module/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_GetTxDataRegisterAddress.
    - Provided independent IRQHandler/transfer APIs for Master and slave to fix the baudrate limit issue.

FLEXIO_I2S

The current FLEXIO_I2S driver version is 2.1.6.

- 2.1.6
  - Bug fix:
    - Added reset flexio before flexio i2s init to make sure flexio status is normal.
- 2.1.5
  - Bug fix:
    - Fixed I2S driver use hard code for bitwidth setting.
- 2.1.4
  - Unified component's 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 disabling module/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.
    - Fixed 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
  - Unified 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 configuration 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
  - Unified 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 configuration 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.

FLEXRAM

The current FLEXRAM driver version is 2.0.5.

- 2.0.5
  - New Features
    - Added the magic address feature for OCRAM, DTCM and ITCM.
- 2.0.4
  - Bug Fixes
    - Fixed FlexRAM driver's missing extern C around functions in header file.
    - Removed magic address feature from driver.
- 2.0.3
  - Bug Fixes
    - Fixed the issue that TCM size configuration was wrong when TCM bank number was not a value power of 2.
- 2.0.2
  - Bug Fixes
    - Updated driver due to Reference Manual update.
- 2.0.1
  - Bug Fixes
    - Fixed MISRA issue.
- 2.0.0
  - Initial version.

FLEXSPI

The current FLEXSPI driver version is 2.1.2.

- 2.1.2
  - Bug Fixes
    - Fixed flag name typos: kFLEXSPI_IpTxFifoWatermarkEmpltyFlag to kFLEXSPI_IpTxFifoWatermarkEmptyFlag; kFLEXSPI_IpCommandExcutionDoneFlag to kFLEXSPI_IpCommandExecutionDoneFlag.
    - Fixed comments typos such as sequencen->sequence, levle->level.
    - Fixed FLSHCR2[ARDSEQID] field clean issue.
    - Updated flexspi_config_t structure and FlexSPI_Init to support new feature FSL_FEATURE_FLEXSPI_HAS_NO_MCR0_ATDFEN and FSL_FEATURE_FLEXSPI_HAS_NO_MCR0_ARDFEN.
    - Updated flexspi_flags_t structure to support new feature FSL_FEATURE_FLEXSPI_HAS_INTEN_AHBBUSERROREN.
- 2.1.1
  - Improvements
    - Defaulted enable prefetch for AHB RX buffer configuration in FLEXSPI_GetDefaultConfig, which is align with the reset value in AHBRXBUFxCR0.
    - Added software workaround for ERR011377 in FLEXSPI_SetFlashConfig; added some

delay after DLL lock status set to ensure correct data read/write.

- 2.1.0
  - New Features
    - Added new API FLEXSPI_UpdateRxSampleClock for users to update read sample clock source after initialization.
    - Added reset peripheral operation in FLEXSPI_Init if required.
- 2.0.5
  - Bug Fixes
    - Fixed FLEXSPI_UpdateLUT cannot do partial update issue.
- 2.0.4
  - Bug Fixes
    - Reset flash size to zero for all ports 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 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
    - Supported DQS write mask enable/disable feature during set FLEXSPI configuration.
    - Provided new API FLEXSPI_TransferUpdateSizeEDMA for users to update eDMA transfer size(SSIZE/DSIZE) per DMA transfer.
  - Bug Fixes
    - Fixed invalid operation of FLEXSPI_Init to enable AHB bus Read Access to IP RX FIFO.
    - Fixed incorrect operation of FLEXSPI_Init to configure IP TX FIFO watermark.
- 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.

GPC

The current GPC driver version is 2.1.1.

- 2.1.1
  - Bug Fix:
    - Move the assert sentence that IRQ register number has to be greater than 0 to platforms which IRQ 0-31 is not available.
- 2.1.0
  - Updated driver for IMX6RT.
- 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 2.0.3.

- 2.0.3 -Bug Fixes
  - MISRA C-2012 issue fixed.
    - Fixed rules, containing: rule-10.3, rule-14.4, and rule-15.5.
- 2.0.2
  - Bug Fixes
    - Fixed the bug of enabling wrong GPIO clock gate in initial API. Since some GPIO instances may not have a clock gate enabled, it checks the clock gate number and makes sure the clock gate is valid.
- 2.0.1
  - Improvements
    - API interface changes:
      - Refined naming of the API while keeping all original APIs, marking them as deprecated. Original APIs 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.

KPP

The current KPP driver version is 2.0.0.

- 2.0.0
  - Initial version.

LPI2C

The current LPI2C driver version is 2.1.10.

- 2.1.10 -Bug Fixes
  - MISRA C-2012 issue fixed.
    - Fixed rule 10.3, 14.4, 15.5.
  - Fixed unaligned access issue in LPI2C_RunTransferStateMachine.
  - Fixed uninitialized variable issue in LPI2C_MasterTransferHandleIRQ.
  - Used linked TCD to disable TX and enable RX in read operation to fix the issue that for platform sharing the same DMA request with TX and RX, during LPI2C read operation if an interrupt with higher priority happened exactly after a command was sent and before TX was disabled, potentially both TX and RX could trigger DMA and cause trouble.
- 2.1.9
  - Bug Fixes
    - Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
    - Fixed Coverity issue of operands did not affect the result in LPI2C_SlaveReceive and LPI2C_SlaveSend.
    - Removed STOP signal wait when NAK detected.
    - Cleared slave repeat start flag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved repeat start flag. This caused the next slave to send a break, and the master was always in the receive data status, but could not receive data.
- 2.1.8
  - Bug Fixes
    - Fixed the transfer issue with LPI2C_MasterTransferNonBlocking, kLPI2C_TransferNoStopFlag, with the wait transfer done through callback in a way of not doing a blocking transfer.
    - Fixed the issue that STOP signal did not appear in the bus when NAK event occurred.
- 2.1.7
  - Bug Fixes
    - Cleared the stopflag before transmission started in LPI2C_SlaveSend/LPI2C_SlaveReceive. The issue was that LPI2C_SlaveSend/LPI2C_SlaveReceive did not handle with the reserved stop flag and caused the next slave to send a break, and the master always stayed in the receive data status but could not receive data.
- 2.1.6
  - Bug Fixes
    - Fixed driver MISRA build error and C++ build error in LPI2C_MasterSend and LPI2C_SlaveSend.
    - Reset FIFO in LPI2C Master Transfer functions to avoid any byte still remaining in FIFO during last transfer.
    - Fixed the issue that LPI2C_MasterStop did not return the correct NAK status in the bus for second transfer to the non-existing slave address.
- 2.1.5
  - Bug Fixes
    - Extended the Driver IRQ handler to support LPI2C4.

    - Changed to use ARRAY_SIZE(kLpi2cBases) instead of FEATURE COUNT to decide the array size for handle pointer array.

- 2.1.4
  - Bug Fixes
    - Fixed the LPI2C_MasterTransferEDMA receive issue when LPI2C shared same request source with TX/RX DMA request. Previously, the API used scatter-gather method, which handled the command transfer first, then the linked TCD which was pre-set with the receive data transfer. The issue was that the TX DMA request and the RX DMA request were both enabled, so when the DMA finished the first command TCD transfer and handled the receive data TCD, the TX DMA request still happened due to empty TX FIFO. The result was that the RX DMA transfer would start without waiting on the expected RX DMA request.
    - Fixed the issue by enabling IntMajor interrupt for the command TCD and checking if there was a linked TCD to disable the TX DMA request in LPI2C_MasterEDMACallback API.
- 2.1.3
  - Improvements
    - 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 Fixes
    - In LPI2C_SlaveTransferHandleIRQ, reset the slave status to idle when stop flag was detected.
- 2.1.1
  - Bug Fixes
    - Disabled the auto-stop feature in eDMA driver. Previously, the auto-stop feature was enabled at transfer when transferring with stop flag. Since transfer was without stop flag and the auto-stop feature was enabled, when starting a new transfer with stop flag, the stop flag would be sent before the new transfer started, causing unsuccessful sending of the start flag, so the transfer could not start.
    - Changed default slave configuration with address stall false.
- 2.1.0
  - Improvements
    - API name changed:
      - 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.

LPSPI

The current LPSPI driver version is 2.0.4.

- 2.0.4
  - Fixed in LPSPI_MasterTransferBlocking that master rxfifo may overflow in stall condition.
- 2.0.3
  - Bug fix:
    - Removed LPSPI_Reset() from LPSPI_MasterInit() and LPSPI_SlaveInit(), because this API may glitch the slave select line. If needed, call this function manually.
- 2.0.2
  - New feature:
    - Added dummy data set up 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 fixes:
    - 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 2.2.8.

- 2.2.8
  - Bug fix:
    - Fixed issue for MISRA-2012 check.
      - Fixed rule-10.3, rule-14.4, rule-15.5.
    - Eliminated Pa082 warnings by assigning volatile variables to local variables and using local variables instead.
  - Improvements:
    - Added check for kLPUART_TransmissionCompleteFlag in LPUART_WriteBlocking, LPUART_TransferHandleIRQ, LPUART_TransferSendDMACallback and LPUART_SendEDMACallback to ensure all the data would be sent out to bus.
    - Rounded up the calculated sbr value in LPUART_SetBaudRate and LPUART_Init to achieve more accurate baudrate setting. Changed osr from uint32_t to uint8_t since osr's biggest value is 31.
- 2.2.7
  - Fixed 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
    - Fixed repeated reading status register issue while dealing with the IRQ routine.
- 2.2.5
    - Bug fix:
        - 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 detecting feature in LPUART_TransferNonBlocking function. If an idle line is 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 ends.
- 2.2.3
    - Changed parameter type in LPUART_RTOS_Init struct from rtos_lpuart_config to lpuart_rtos_config_t.
    - Bug fix:
        - Disabled LPUART receive interrupt instead of all NVIC when reading data from ring buffer. Otherwise when the ring buffer is used, receive nonblocking method will disable all NVICs to protect the ring buffer. This may has a negative effect on other IPs that 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 unnecessary check of event flags and assert in LPUART_RTOS_Receive.
    - Always wait for RX event flag in LPUART_RTOS_Receive.
- 2.1.0
    - Update transactional APIs.

PIT

The current PIT driver version is 2.0.2.

- 2.0.2

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

PMU

The current PMU driver version is 2.1.0.

- 2.1.0
  - Improvements
    - Added feature macros for low power control APIs to support conditional compile.
    - Renamed "PMU_2P1EnablePullDown" to "PMU_2P5EnablePullDown".
- 2.0.0
  - Initial version.

PWM

The current PWM driver version is 2.0.0.

- 2.0.0
  - Initial version.

RTWDOG

The current RTWDOG driver version is 2.1.0.

- 2.1.0
  - Added API to enable or disable the window mode.
  - Added API to convert a raw count value to millisecond.
  - Used AT_QUICKACCESS_SECTION_CODE macro to decorate RTWDOG_Init, copy this function from flash to QUICKACCESS section.
- 2.0.1
  - Fixed bug in the RTWDOG_Init, added check of register's unlock status when configuring the RTWDOG in RTWDOG_init.
- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.2.2.

-2.2.2 Bug fix:

- Fix the issue of MISRA 2004 rule 9.3.

2.2.1

- Improvements:
  - Added mclk post divider support in function SAI_SetMasterClockDivider.
  - Removed useless configuration code in SAI_RxSetSerialDataConfig.
- Bug fix:
  - Fixed the SAI SDMA driver build issue caused by the wrong structure member name used in the function SAI_TransferRxSetConfigSDMA/SAI_TransferTxSetConfigSDMA.
  - Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CHANNEL_COUNTn.
  - Apply ERR05144: not set FCONT = 1 when TMR > 0, otherwise the TX may not work.

2.2.0

- Improvements:
  - 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 SAI_GetTDMConfig

2.1.9

- Improvements:
  - Improved SAI driver comment for clock polarity.
  - Added enumeration for SAI for sample inputs on different edge.
  - Changed FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHANNEL_COUNTn(base) for the difference between the different SAI instance.
  - Added new API: SAI_TxSetBitClockDirection SAI_RxSetBitClockDirection SAI_RxSetFrameSyncDirection SAI_TxSetFrameSyncDirection

2.1.8

- Improvements:
  - Added feature macro test for the sync mode2 and mode 3.
  - Added feature macro test for masterClockHz in sai_transfer_format_t.

2.1.7

- Improvements:
  - Added feature macro test for the mclkSource member in sai_config_t.

  - Changed "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:
  - Fixed build error caused by feature macro test for mclkSource.

-2.1.6

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

-2.1.5

- Bug fix:
  - Corrected SAI3 driver IRQ handler name.
  - Added I2S4/5/6 IRQ handler.
  - Added base in handler structure to support different instances share one IRQ number.
- New features:
  - Updated SAI driver for MCR bit MICS.
  - Added 192 KHZ/384 KHZ 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.

SPDIF

The current SPDIF driver version is 2.0.2.

- 2.0.2 Bug fix:
  - Corrected operator used for size value assertion in SPDIF_ReadBlocking/SPDIF_WriteBlocking.
- 2.0.1
  - Corrected the feature macro name used to define s_edmaPrivateHandle.
- 2.0.0
  - Initial version.

SRC

The current SRC driver version is 2.0.1.

- 2.0.1
  - Improvements: Updated SRC driver for adding SRC_SRSR_JTAG_SW_RST enumeration.
- 2.0.0
  - Initial version.

TEMPMON

The current TEMPMON driver version is 2.0.0.

- 2.0.0
  - Initial version.

WDOG

The current WDOG driver version is 2.1.0.

- 2.1.0
  - New Features
    - Added new API "WDOG_TriggerSystemSoftwareReset()" to allow users to reset the system by software.
    - Added new API "WDOG_TriggerSoftwareSignal()" to allow users to trigger a WDOG_B signal by software.
    - Removed the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog_config_t structure.
    - Added new parameter "enableTimeOutAssert" to the wdog_config_t structure. With this parameter enabled, when the WDOG timeout occurs, a WDOG_B signal will be asserted. This signal can be routed to external pin of the chip. Note that WDOG_B signal remains asserted until a power-on reset (POR) occurs.
- 2.0.1
  - New Features
    - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

XBARA

The current XBARA driver version is 2.0.4.

- 2.0.4
  - Improvement:
    - Optimized XBARA_SetOutputSignalConfig.
- 2.0.3
  - Bug fixes:
    - Corrected configuration for function XBAR_SetOutputSignalConfig.
- 2.0.2
  - Other changes:
    - Changed array clock name.
- 2.0.1
  - Bug fixes:
    - Fixed w1c bits for XBARA_SetOutputSignalConfig function.
- 2.0.0
  - Initial version.

2 Middleware Change Log

FatFs for MCUXpresso SDK

Current version is FatFs R0.13c_rev0.

- R0.13c_rev0
  - Upgraded to version 0.13c
  - Apply patches ff_13c_p1.diff,ff_13c_p2.diff, ff_13c_p3.diff and ff_13c_p4.diff.
- 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.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.4.2.

- 2.4.2
  - Improvement:
    - Put the USB controller data and transfer buffer to noncache section, removed the setting that sets the whole OCRAM and SDRAM as noncached.
    - Separated composite audio examples' channel,sample rate,format parameters from common macro to in-dedicated macro and out-dedicated macro.
    - replaced USB_PrepareData with USB_AudioRecorderGetBuffer.
- 2.4.1
  - New features:
    - Added enumeration fail callback to host stack when the attached device's enumeration failed.
- 2.4.0
  - Improvement:

        - Device Charger Detection (DCD) software architecture was refactored.
    - New features:
        - Enabled Device Charger Detection (DCD) on RT1060.
        - Enabled Device Charger Detection on RT600.
        - Enabled host battery charger function on RT600.
- 2.3.0
    - New features:
        - Added host video camera support. example: usb_host_video_camera
        - Added a new device example. example: usb_device_composite_cdc_hid_audio_unified
- 2.2.0
    - New features:
        - Added device DFU support.
        - Supported OM13790DOCK on LPCXpresso54018.
        - Added multiple logical unit support in msc class driver, updated usb_device_lba_-information_struct_t to support this.
        - Supported multiple transfers for host ISO on IP3516HS.
    - Bug fixes:
        - Fixed device ip3511 prime data length than maxpacket size issue.
        - Initialized interval attribute in usb_device_endpoint_struct_t/usb_device_endpoint_init_-struct_t.
        - Removed unnecessary header file in device CDC class driver, removed unnecessary usb_-echo, and added DEBUG macro for necessary usb_echo in device CDC class driver.
        - Fixed device IP3511HS unfinished interrupt transfer missing issue.
- 2.1.0
    - New features:
        - Added host RNDIS support. example: lwip_dhcp_usb
        - Enabled USB 3.0 support on device stack.
        - Power Delivery feature: Added OM13790HOST support; Added auto policy feature; Printed e-marked cable information;
- 2.0.1
    - Bug fixes:
        - Fixed some USB issues: Fixed MSC CV test failed in MSC examples.
        - Changed audio codec interfaces.
- 2.0.0
    - New features:
        - PTN5110N support.
    - Bug fix:
        - Added some comments, fixed some minor USB issues.
- 1.9.0
    - New features:
        - Examples:
            - usb_pd_alt_mode_dp_host
- 1.8.2
    - Updated license.
- 1.8.1

    - Bug fix:
        - Verified some hardware issues, support aruba_flashless.
- 1.8.0
    - New features:
        - Examples:
            - usb_device_composite_cdc_vcom_cdc_vcom
            - usb_device_composite_hid_audio_unified
            - usb_pd_sink_battery
            - Changed usb_pd_battery to usb_pd_charger_battery.
    - Bug fix:
        - Code clean up, removed some irrelevant code.
- 1.7.0
    - New features:
        - USB PD stack support.
    - Examples:
        - usb_pd
        - usb_pd_battery
        - usb_pd_source_charger
- 1.6.3
    - Bug fix: -IP3511_HS driver control transfer sequence issue, enabled 3511 ip cv test.
- 1.6.2
    - New features:
        - Multi instance support.
- 1.6.1
    - New features:
    - Changed the struct variable address method for device_video_virtual_camera and host_phdc_manager.
- 1.6.0
    - New features:
        - Supported Device Charger Detect feature on usb_device_hid_mouse.
- 1.5.0
    - New features:
        - Supported controllers
            - OHCI (Full Speed, Host mode)
            - IP3516 (High Speed, Host mode)
            - IP3511 (High Speed, Device mode)
        - Examples:
            - usb_lpm_device_hid_mouse
            - usb_lpm_device_hid_mouse_lite
            - usb_lpm_host_hid_mouse
- 1.4.0
    - New features:
        - Examples:
            - usb_device_hid_mouse/freertos_static
            - usb_suspend_resume_device_hid_mouse_lite

- 1.3.0
  - New features:
    - Supported roles
      - OTG
    - Supported classes
      - CDC RNDIS
    - Examples
      - usb_otg_hid_mouse
      - usb_device_cdc_vnic
      - usb_suspend_resume_device_hid_mouse
      - usb_suspend_resume_host_hid_mouse
- 1.2.0
  - New features:
    - Supported controllers
      - LPC IP3511 (Full Speed, Device mode)
- 1.1.0
  - Bug fix:
    - Fixed some issues in USB certification.
    - Changed VID and Manufacturer string to NXP.
  - New features:
    - Supported classes
      - Pinter
    - Examples:
      - usb_device_composite_cdc_msc_sdcard
      - usb_device_printer_virtual_plain_text
      - usb_host_printer_plain_text
- 1.0.1
  - Bug fix:
    - Improved the efficiency of device audio speaker by changing the transfer mode from interrupt to DMA, thus providing the ability to eliminate the periodic noise.
- 1.0.0
  - New features:
    - Supported roles
      - Device
      - Host
    - Supported controllers:
      - KHCI (Full Speed)
      - EHCI (High Speed)
    - Supported classes:
      - AUDIO
      - CCID
      - CDC
      - HID
      - MSC
      - PHDC

        - VIDEO
    - Examples:
        - usb_device_audio_generator
        - usb_device_audio_speaker
        - usb_device_ccid_smart_card
        - usb_device_cdc_vcom
        - usb_device_cdc_vnic
        - usb_device_composite_cdc_msc
        - usb_device_composite_hid_audio
        - usb_device_composite_hid_mouse_hid_keyboard
        - usb_device_hid_generic
        - usb_device_hid_mouse
        - usb_device_msc_ramdisk
        - usb_device_msc_sdcard
        - usb_device_phdc_weighscale
        - usb_device_video_flexio_ov7670
        - usb_device_video_virtual_camera
        - usb_host_audio_speaker
        - usb_host_cdc
        - usb_host_hid_generic
        - usb_host_hid_mouse
        - usb_host_hid_mouse_keyboard
        - usb_host_msd_command
        - usb_host_msd_fatfs
        - usb_host_phdc_manager
        - usb_keyboard2mouse
        - usb_pin_detect_hid_mouse

QCA WiFi

The current version is 2.0.0.

- 2.0.0
    - Initial version.
        - Added QCA WiFi, ported from SDK 1.3, synchronized with latest MQX Qualcomm v3.-3.5.
    - Known issues:
        - Low power mode may not work, require further investigation.
        - DHCP request requires some timeout to retrieve valid data.

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_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/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.

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

Date of release: 10/2019
Document identifier: MCUXSDKMIMXRT1011RN

arm