NXP Semiconductors
Release Notes

Document Number: MCUXSDKIMX8MMRN
Rev. 0, 01/2019

MCUXpresso SDK Release Notes for i.MX 8M Mini

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 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

Freescale prefix 'FSL' have 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 8.30.2
- Makefiles support with GCC revision v7-2017-q4-major from Arm Embedded

4 Supported development systems

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

Table 1. Supported devices and development boards

Development boards	Devices
MIMX8MM EVK board	**MIMX8MM6DVTLZ**, MIMX8MM6CVTKZ, MIMX8MM5DVTLZ, MIMX8MM5CVTKZ, MIMX8MM4DVTLZ, MIMX8MM4CVTKZ, MIMX8MM3DVTLZ, MIMX8MM3CVTKZ, MIMX8MM2DVTLZ, MIMX8MM2CVTKZ, MIMX8MM1DVTLZ, MIMX8MM1CVTKZ

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
Multicore examples	<install_dir>/boards/<board_name>/multicore_examples
Documentation	<install_dir>/docs
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Multicore stack	<install_dir>/middleware/multicore
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

Table continues on the next page...

Table 2. Release contents (continued)

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.
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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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.
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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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.

MCUXpresso SDK Release Notes for i.MX 8M Mini

Change Logs

Contents

1 Driver Change Log

ECSPI

The current eCSPI driver version is 2.0.1.

- 2.0.1
  - Memset local variable sdma xfer config structure to make sure unused members in structure are cleared.
- 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.

I2C

The current I2C driver version is 2.0.4.

- 2.0.4
  - Bug fix:
    - Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) does not support the situation that master transfer with subaddress and transfer data size zero, which means no data follows by the subaddress.
- 2.0.3
  - Improvement:
    - Improved code readability, add new static API I2C_WaitForStatusReady for the status flag

wait, and change to call I2C_WaitForStatusReady instead of polling flags with reading register.

- 2.0.2
  - Improvement:
    - Added I2C_WATI_TIMEOUT macro to allow user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.1
  - Bug fix:
    - Added proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Only supports write only or write+read with no start flag, does not support read only with no start flag.
- 2.0.0
  - Initial version.

TMU

The current TMU driver version is 2.0.0.

- 2.0.0
  - Initial version.
  - This module is developed firstly on i.MX 8MM.

PDM

The current PDM driver version is 2.0.1.

- 2.0.1
  - Improve HWVAD feature.
- 2.0.0
  - Initial version.

PWM

The current PWM driver version is 2.0.0.

- 2.0.0
  - Initial version.

UART

The current UART driver version is 2.0.1.

- 2.0.1

  - Memset local variable sdma xfer config structure to make sure unused members in structure are cleared.
- 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.

RDC

The current RDC driver version is 2.0.0.

- 2.0.0
  - Initial version.

RDC_SEMA42

The current RDC_SEMA42 driver version is 2.0.0.

- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.1.7.

- 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_Rx-

Init.

2.0.0

- Initial version.

SDMA

The current SDMA driver version is 2.1.0.

- 2.1.0
  - Add SDMA_SetMultiFifoConfig api to support multi fifo feature.
  - Add SDMA_EnableSwDone api to support sw done feature.
  - Add SDMA_LoadScript api to support load script to sdma program memory.
  - Add SDMA_DumpScript api to support dump script from sdma program memory.
  - Add SDMA3 IRQ handler.
- 2.0.0
  - Initial version.

SEMA4

The current SEMA4 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 Feature:
    * Add new API "WDOG_TriggerSystemSoftwareReset()" to allow user reset the system by software.
    * Add new API "WDOG_TriggerSoftwareSignal()" to allow user trigger a WDOG_B signal by software.
    * Remove the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog_config_t structure.
    * Add new parameter "enableTimeOutAssert" to the wdog_config_t structure, with this parameter enabled, while the WDOG timeout occurred, a WDOG_B signal will be asserted, this signal can be routed to external pin of the chip. Please note that, WDOG_B signal remains asserted until a power-on reset (POR) occurs.
- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0

- Initial version.

2 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
  - New features:
    - Support for CM33, CM33F architectures based on CM3, CM4F ports
    - 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

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.

Document Number MCUXSDKIMX8MMRN
Revision 0, 01/2019