

MCUXpresso SDK Release Notes Supporting EVK-MIMXRT1020



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.....	9
Chapter 7 MISRA compliance.....	12
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 FatFs, USB, lwIP, mbed TLS cryptography libraries, other middleware packages,, and integrated RTOS support for FreeRTOS™ OS. In addition to the base enablement, the MCUXpresso SDK is augmented with demo applications, 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](#).

Chapter 2

MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpresso SDK is the evolution of Kinetis SDK v2.7.0, includes support for both LPC and i.MX System-on-Chips (SoC). The same drivers, APIs, and middleware are still available with support for Kinetis, LPC, and i.MX silicon. The MCUXpresso SDK adds support for the MCUXpresso IDE, 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)® 5.29
- Makefiles support with GCC revision 8-2019-q3 GCC8 from Arm Embedded
- MCUXpresso IDE v11.1.0

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
EVK-MIMXRT1020	MIMXRT1021CAF4A, MIMXRT1021CAG4A, MIMXRT1021DAF5A, MIMXRT1021DAG5A

Chapter 5

Release contents

Table 2 provides an overview of the MCUXpresso SDK release package contents and locations.

Table 2. Release contents

Deliverable	Location
AWS Device Configuration	<install_dir>/boards/<board>/aws_examples/ device_configuration_android/AwsDeviceConfiguration.apk
AWS IoT	<install_dir>/middleware/aws_iot
AWS IoT SDK examples	<install_dir>/boards/<board_name>/aws_examples
AWS LED WiFi	<install_dir>/boards/<board>/aws_examples/led_wifi_android/ AwsLedWifi.apk
AWS Remote Control	<install_dir>/boards/<board>/aws_examples/remote_control_android/ AwsRemoteControl.apk
Boards	<install_dir>/boards
CMSIS Arm Cortex [®] -M header files, DSP library source	<install_dir>/CMSIS
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Cortex Microcontroller Software Interface Standard (CMSIS) driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
Demo applications	<install_dir>/boards/<board_name>/demo_apps
Documentation	<install_dir>/docs
Driver examples	<install_dir>/boards/<board_name>/driver_examples
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
FatFS stack	<install_dir>/middleware/fatfs
FNET	<install_dir>/boards/<board>/aws_examples/ device_configuration_enet/fnet_mdns
gradle	<install_dir>/boards/<board>/aws_examples/remote_control_android/ gradle, boards/<board>/aws_examples/led_wifi_android/gradle, boards/<board>/aws_examples/device_configuration_android/gradle
jsmn	<install_dir>/middleware/aws_iot/external_libs/jsmn
LwIP demo applications	<install_dir>/boards/<board_name>/lwip_examples
LwIP Documentation	<install_dir>/docs/lwip
LwIP stack	<install_dir>/middleware/lwip
mbed TLS	<install_dir>/middleware/mbedtls
mbed TLS examples	<install_dir>/boards/<board_name>/mbedtls_examples

Table continues on the next page...

Table 2. Release contents (continued)

Deliverable	Location
percepio_snapshot	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_percepio_snapshot
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
Qualcomm WiFi	<install_dir>/middleware/wifi_qca
Qualcomm WiFi stack examples	<install_dir>/boards/<board_name>/wifi_qca_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
RTOS Kernel Code	<install_dir>/rtos
segger_systemview	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_segger_sysview
Tools	<install_dir>/tools
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
USB stack	<install_dir>/middleware/usb
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities

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

[Table 3](#) provides a list of USB devices tested with the USB Host stack.

Table 3. Peripheral devices

Device type	Device
USB HUB	BELKIN F5U233
	BELKIN F5U304
	BELKIN F5U307
	BELKIN F4U040
	UNITEK Y-2151

Table continues on the next page...

Table 3. Peripheral devices (continued)

Device type	Device
	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/adaptor	SSK TF adapter Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card
USB Mouse	DELL MS111-P DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
USB Keyboard	DELL SK8135 DELL SK8115

6.2.2 TCP/IP stack

The lwIP TCP/IP stack is pre-integrated with MCUXpresso SDK and runs on top of the MCUXpresso SDK Ethernet driver with Ethernet-capable devices/boards.

6.2.3 File system

The FatFs file system is integrated with the MCUXpresso SDK and can be used to access either the SD card or the USB memory stick when the SD card driver or the USB Mass Storage Device class implementation is used.

6.2.4 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.5 CMSIS

The MCUXpresso SDK is shipped with the standard CMSIS development pack, including the prebuilt libraries.

Chapter 7

MISRA compliance

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

Table 4. MISRA exceptions

Exception Rules	Description
Rule 5.1	External identifiers shall be distinct.
Rule 5.4	Macro identifiers shall be distinct.
Rule 21.1	#define and #undef shall not be used on a reserved identifier or reserved macro name.
Rule 21.2	A reserved identifier or macro name shall not be declared.
Directive 4.4	Sections of code should not be "commented out".
Directive 4.5	Identifiers in the same name space with overlapping visibility should be typographically unambiguous.
Directive 4.6	Typedefs that indicate size and signedness should be used in place of the basic numerical types.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden.
Directive 4.9	A function should be used in preference to a function-like macro where they are interchangeable.
Directive 4.13	Functions which are designed to provide operations on a resource should be called in an appropriate sequence.
Rule 1.2	Language extensions should not be used.
Rule 2.3	A project should not contain unused type declarations.
Rule 2.4	A project should not contain unused tag declarations.
Rule 2.5	A project should not contain unused macro declarations.
Rule 2.6	A function should not contain unused label declarations.
Rule 2.7	There should be no unused parameters in functions.
Rule 4.2	Trigraphs should not be used.
Rule 5.9	Identifiers that define objects or functions with internal linkage should be unique.
Rule 8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.
Rule 8.9	An object should be defined at block scope if its identifier only appears in a single function.
Rule 8.11	When an array with external linkage is declared, its size should be explicitly specified.

Table continues on the next page...

Table 4. MISRA exceptions (continued)

Rule 8.13	A pointer should point to a const-qualified type whenever possible.
Rule 10.5	The value of an expression should not be cast to an inappropriate essential type.
Rule 11.4	A conversion should not be performed between a pointer to object and an integer type.
Rule 11.5	A conversion should not be performed from pointer to void into pointer to object.
Rule 12.1	The precedence of operators within expressions should be made explicit.
Rule 12.3	The comma operator should not be used.
Rule 12.4	Evaluation of constant expressions should not lead to unsigned integer wrap-around.
Rule 13.3	A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator.
Rule 15.4	There should be no more than one break or go to statement used to terminate any iteration statement.
Rule 17.5	The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements.
Rule 17.8	A function parameter should not be modified.
Rule 19.2	The union keyword should not be used.
Rule 20.1	#include directives should only be preceded by preprocessor directives or comments.
Rule 20.10	The #and ## preprocessor operators should not be used.
Rule 21.12	The exception handling features of <fenv.h> should not be used. .

Chapter 8

Known issues

8.1 Maximum file path length in Windows 7[®] operating system

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`.

8.4 CMSIS PACK new project compile failure

The generated configuration cannot be applied globally. The components, `serial_manager_usb_cdc_virtual` and `serial_manager_usb_cdc_virtual_xxx` ("xxx" means core variants like `cm0plus`, `cm33`, `cm4`, `cm33_nodsp`) are unsupported for new project wizard of CMSIS pack and will lead to compile failure if selected while creating a new project(s).

8.5 RAM targets build issue in CMSIS bsp pack

Because CMSIS pack does not support different macro definitions for different targets, all RAM targets for projects inside CMSIS BSP PACKs for RT10XX boards will get the same macro definitions with Flash targets, resulting in build failure. To pass build for RAM targets, manually update the `XIP_EXTERNAL_FLASH` and `XIP_BOOT_HEADER_ENABLE` value to 0 in `RTE_Components.h`.

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, μ Vision, 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.

© NXP B.V. 2019.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 12/2019

Document Identifier: MCUXSDKMIMXRT102XRN

