MCUXSDKMIMXRT106XRN

MCUXpresso SDK Release Notes for EVK-MIMXRT1060

Rev. 0 — 14 July 2020

Contents

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.

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpresso SDK is the evolution of Kinetis SDK, 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 the 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. It is suggested to keep the downloaded SDK archive in the root directory of your drive to avoid any unexpected build issues caused by deep path of files.

3 Development tools

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

- IAR Embedded Workbench for Arm version 8.50.5
- MDK-Arm Microcontroller Development Kit (Keil)® 5.31
- Makefiles support with GCC revision 9-2019-q4-major GCC9 from Arm Embedded
- MCUXpresso IDE v11.2.0

4 Supported development systems

This release supports boards and devices listed in Table 1. The boards and devices in bold were tested in this release.

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
EVK-MIMXRT1060	MIMXRT1062CVJ5A,MIMXRT1062CVL5A, MIMXRT1062DVJ6A, **MIMXRT1062DVL6A**, MIMXRT1061CVJ5A, MIMXRT1061CVL5A, MIMXRT1061DVJ6A, MIMXRT1061DVL6A

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 SDK examples	`<install_dir>/boards/<board_name>/aws_examples`
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`
Cypress Wiced SDK (WiFi, BLE)	`<install_dir>/middleware/wiced`
Cypress WiFi stack examples	`<install_dir>/boards/<board_name>/wifi_cypress_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>`
emWin examples	`<install_dir>/boards/<board_name>/emwin_examples`
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`
LittleFS examples	`<install_dir>/boards/<board_name>/littlefs_examples`
LittleFS	`<install_dir>/middleware/littlefs`

Table continues on the next page...

Table 2. Release contents (continued)

Deliverable	Location
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
NXP WiFi	<install_dir>/middleware/wifi
NXP WiFi examples	<install_dir>/boards/<board_name>/wifi_examples
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
Secure element host library examples	<install_dir>/boards/<board_name>/se_hostlib_examples
TinyCBOR	<install_dir>/rtos/amazon-freertos/lib/third_party/tinycbor
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

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 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

Table continues on the next page...

Table 3. Peripheral devices (continued)

Device type	Device
	SAMSUNG USB3.0 16GB
USB card reader/adapter	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 NXP WiFi drivers

The MCUXpresso SDK provides driver for NXP WiFi external modules. The WiFi driver is integrated with LWIP TCPIP stack and demonstrated with several network applications (iperf and AWS IoT).

The following are the supported external processor and modules:

- NXP 88W8977 based AW-AM281SM AzureWave module (M.2 Kit : AW-AM281MA)
- NXP 88W8801 based AW-NM191SM AzureWave module (M.2 Kit : AW-AM191NM)

6.2.4 Cypress Wiced SDK (WiFi, Bluetooth Low Energy)

The MCUXpresso SDK provides integration with Cypress Wiced SDK supporting the Murata Type 1DX and Azurewave AW-NM372SM modules based on the CYW4343W and CYW43438 processors.

6.2.5 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.

6.2.6 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.7 CMSIS

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

6.2.8 emWin

The MCUXpresso SDK is pre-integrated with the SEGGER emWin GUIBuilder.

7 MISRA compliance

All MCUXpresso SDK drivers comply to MISRA 2012 rules with exceptions in Table 4.

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	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 5.1	External identifiers shall be distinct.
Rule 5.4	Macro identifiers shall be distinct.
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)

Exception rules	Description
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 21.1	`#define` and `#undef` shall not be used on a reserved identifieror reserved macro name.
Rule 21.2	A reserved identifier or macro name shall not be declared.
Rule 21.12	The exception handling features of `<fenv.h>` should not be used.

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 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, such as, `cm0plus`, `cm33`, `cm4`, `cm33_nodsp`.

Components: `serial_manager_uart`, `serial_manager_uart_xxx`, `serial_manager_usb_cdc`, `serial_manager_usb_cdc_xxx`, `serial_manager_swo`, `serial_manager_swo_xxx`, `xip_device`.

Also for low-level adapter components, currently the different types of the same adapter cannot be selected at the same time. For example, if there are two types of timer adapters, `gpt_adapter` and `pit_adapter`, only one can be selected as timer adapter in one project at a time. Duplicate implementation of the function results in an error.

NOTE

Most of middleware components have complex dependencies and are not fully supported in new project wizard. Adding a middleware component may result in compile failure.

8.3 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.

8.4 Invalid path for "-imacros" argument in project with linked sources in MCUXpresso IDE

This issue is related to examples using wifi middleware (wifi_iperf, aws_*_wifi_nxp).

Examples imported with unchecked option "Copy sources" have set invalid path to wifi_config.h in argument "-imacros".

Update the Properties of imported project:

C/C++ Build > Settings > ToolSettings Tab > MCU C Compiler > Miscellaneous > Other flags:

```
-imacros "${ProjDirPath}/source/wifi_config.h"
```

change to:

```
-imacros "wifi_config.h"
```

8.5 Keil MDK high compiler optimization issue (wifi_iperf, aws_shadow_wifi_nxp)

The following examples does not work correctly in Keil MDK toolchain on high optimization setting (flexspi_nor_release target)

- wifi_iperf - this issue can be fixed by changing optimization from “-Oz” to “-Os” in project settings
- aws_shadow_wifi_nxp - this issue can be fixed by changing optimization from “-Oz” to “-O0” in project settings

Decrease optimization level in Arm MDK, in case NXP wifi driver is used.

8.6 Non XIP target debug issue on toolchain MDK

When debugging non XIP targets in flash boot mode, if application changes any settings which have impacts on flexspi, the build output window might show "Debug access failed" when start debugging next time. It is recommended to keep the board in serial downloader mode when debugging non XIP targets.

8.7 Corrupted data in freertos_lpspi_b2b (slave) example

Corrupted data in freertos_lpspi_b2b(slave) example.

Tool: Keil MDK

Target: freertos_lpspi_b2b_slave_flexspi_nor_debug.

Changing the optimization level from -01 to -00 can avoid the issue. However, the optimization level -O1 is not the root cause.

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, 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, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

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

Date of release: 14 July 2020
Document identifier: MCUXSDKMIMXRT106XRN

arm