NXP Semiconductors
User's Guide

Document identifier: MCUXSDKK66FRN
Rev. 0, 12/2019

MCUXpresso SDK Release Notes Supporting TWR-K65F180M and FRDM-K66F

Contents

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, mbed TLS cryptography libraries, other middleware packages,, and integrated RTOS support for FreeRTOS^{TM} 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
TWR-K65F180M, FRDM-K66F	MK65FN2M0VMI18WS, MK65FN2M0CAC18WS, MK65FX1M0CAC18WS, MK65FX1M0VMI18WS, MK26FN2M0VLQ18, MK26FN2M0VMD18, MK26FN2M0CAC18, MK26FN2M0VMI18, **MK65FN2M0VMI18**, MK65FN2M0CAC18, MK65FX1M0CAC18, MK65FX1M0VMI18, **MK66FN2M0VMD18**, MK66FN2M0VLQ18, MK66FX1M0VMD18, MK66FX1M0VLQ18

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
Azure IoT	<install_dir>/middleware/azure_iot
Azure IoT SDK examples	<install_dir>/boards/<board_name>/azure_examples
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>
emWin examples	<install_dir>/boards/<board_name>/emwin_examples
FatFS stack	<install_dir>/middleware/fatfs
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
mbed TLS	<install_dir>/middleware/mbedtls
mbed TLS examples	<install_dir>/boards/<board_name>/mbedtls_examples
mmCAU	<install_dir>/middleware/mmcau
mmCAU examples	<install_dir>/middleware/mmcau_examples

Table continues on the next page...

Table 2. Release contents (continued)

Deliverable	Location
Multiprocessor examples	`<install_dir>/boards/<board_name>/multiprocessor_examples`
percepio_snapshot	`<install_dir>/boards/<board>/rtos_examples/visualization/freertos_percepio_snapshot`
Peripheral Drivers	`<install_dir>/devices/<device_name>/drivers`
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`
TinyCBOR	`<install_dir>/rtos/amazon-freertos/lib/third_party/tinycbor`
Tools	`<install_dir>/tools`
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 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.

6.2.2 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.

6.2.4 emWin

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

Chapter 7
MISRA compliance

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

Table 3. 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 identifieror 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 3. 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 USB HUB power supply

The external power supply of the USB HUB must be provided before it can be used. The development board power is not enough to supply multi-level USB HUBs and connected devices. Therefore, the external USB HUB that is connected to the development board should have its own power supply.

8.3 USB PID issue

Because the PID of all USB device examples is updated, uninstall the device drivers and then reinstall when the device (with new PID) is plugged in the first time.

8.4 No sound issue

Under EHCI of the TWRK65F180M, if the USB device audio speaker connects to a high-speed hub along with an HID device, such as a keyboard or mouse, there is high possibility the sound will not work.

8.5 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).

8.6 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.

8.7 CMSIS PACK new project compile failure

The generated configration 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).

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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: MCUXSDKK66FRN

arm