

MCUXpresso SDK Release Notes for JN518x



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..... 10

Chapter 8 Known issues..... 12

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 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](#).

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 v2.7.1, 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 8.32.4
- MCUXpresso IDE v11.1.0
- Python 2.7 (Used by IAR and MCUXpresso IDE post build script)
- GUN Tools for Arm Embedded Processors 8-2018-q4-major

Chapter 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
OM15076-V3 (DK006) Carrier Board + JN518x module	JN5189, JN5189T, JN5188, JN5188T
JN5189-USB Dongle	JN5189

Chapter 5

Release contents

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

Table 2. Release contents

Deliverable	Location
Boards	<install_dir>/boards
CMSIS Arm Cortex® -M header files, DSP library source	<install_dir>/CMSIS
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>
Middleware	<install_dir>/middleware
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools
USB Dongle Examples	<install_dir>/boards/<board_name>/usb_dongle_examples
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
Wireless examples	<install_dir>/boards/<board_name>/wireless_examples

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, 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.

6.2 Middleware

6.2.1 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.2 OpenThread

This version corresponds to the Release for Production (RFP) build of the JN5189DK6 Software.

The main features of this release are listed below.

- The OpenThread stack implements version 1.1.1 of the Thread Group core specification.
- The OpenThread stack comes with application examples for implementing Thread Router Eligible Device, Thread End Device (including Low Power/Sleepy End Devices), and Thread Border Router, including application examples for implementing the Over-The-Air Updates in a Thread Large Network.

6.2.3 Zigbee 3.0

ZigBee Pro stack, Base Device Behavior, and ZigBee cluster library are implemented in line with the ZigBee 3.0 version of the standards.

ZigBee Compliant Platform is certified to the ZigBee Pro R22 specifications.

The SDK is supplied with ZigBee 3.0 with wireless demo examples for implementing a ZigBee Coordinator, ZigBee Router, Zigbee sleepy (Rx OFF when Idle) End Device and Zigbee none-sleeping (Rx ON) End Device, with the option for Over-the-Air updates.

Additional reference ZigBee Application Notes for ZigBee ZLO devices such as Bulbs, Controller & Switches, Sensors and Control Bridge are distributed outside of the SDK on the NXP website.

6.2.4 Other middleware

Optional middleware packages can be included in the release based on the user selection. See `<install_dir>/SW-Content-Register.txt` for a list of components and associated licenses.

Chapter 7

MISRA compliance

All MCUXpresso SDK drivers comply to MISRA 2012 rules with exceptions in [Table 3](#).

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	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.
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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

Exception rules	Description
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	<code>#include</code> directives should only be preceded by preprocessor directives or comments.
Rule 20.10	The <code>#</code> and <code>##</code> preprocessor operators should not be used.
Rule 21.1	<code>#define</code> and <code>#undef</code> 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.
Rule 21.12	The exception handling features of <code><fenv.h></code> 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 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: 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.

8.3 Other limitations

- The OpenThread Large Network has been tested for 110 nodes.
- The OpenThread Border Router application is natively available for RCP configuration and demonstrated in this release. NCP configuration is not demonstrated in current release.
- There are times when the Thread Commissioning App cannot discover/connect to the mDNS services exposed by the Border Router. In this case, please connect to the UDOO shell and restart the web agent service by running the `restart_br_agent.sh` bash script (`sudo ./restart_br_agent.sh`). The role of this script is to connect to the running Docker container and restart the otr-agent service.
- After an OTA upgrade, a reflash of the board from IAR is possible only if the NVM is erased.
- If a JN518x device is erased and left without an image in flash, the MCUXpresso IDE will be unable to access it for programming or debug. The workaround is to program any image into the device using either the IAR IDE or the DK6 CLI Flash Programmer. An erase-program performed as a single operation from within the MCUXpresso IDE works correctly.
- There is no RTOS support in this release. The Zigbee application is designed to work as a bare metal implementation.
- The OpenThread and ZigBee wireless demo applications contained within this release are designed to operate as bare metal implementations only. There is currently no RTOS support for these wireless applications in this release.
- SW4/DIO4 on OM15082 should not be used as mask for Wake.
- The PSECT (protected sectors of flash) is not in the normal application flash use domain. It is used to store certain chip related configurations. It is possible to write to these pages through the flash programmer.

NOTE

DO NOT write to these locations. Writing to these sectors outside their intended use will cause the device to not function correctly.

- IAR toolchain is not currently supported. Only MCUXpresso is supported for all the Zigbee Wireless applications.
MAC OS and Linux are not supported with Zigbee.
- The binary generated by the IAR Output Converter should not be run on the chip since it does not contain the signature information. The correct one is `xxxx.out.bin` in the debug or release directory of the project.

8.4 Need to use QN9090 as the part number in IAR project file

IAR does not officially support the JN5189 yet. Please use QN9090 as the part number in IAR project file. All SDK examples are using the QN9090 part number in IAR project file.

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. 2020.

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: 03/2020

Document identifier: MCUXSDKJN518XRN