Document Number: MCUXSDKLPC8XXRN
Rev. 1, 06/2018

MCUXpresso SDK Release Notes Supporting LPCXpresso8XX

Contents

1 Overview....1
2 MCUXpresso SDK....1
3 Development tools....2
4 Supported development systems....2
5 Release contents....2
6 MCUXpresso SDK release package....3
7 MISRA compliance....4
8 Known issues....6
9 Revision history....7

1 Overview

The MCUXpresso Software Development Kit (SDK) is a collection of software enablement for Microcontrollers that includes peripheral drivers, high-level stacks including USB and lwIP, integration with WolfSSL and mbed TLS cryptography libraries, other middleware packages, such as multicore support and FatFs, 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.

For the latest version of this and other MCUXpresso SDK documents, 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.

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, MCUXpressoSDK is the evolution of Kinetis SDK v2.3.0, 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, a new Eclipse-based toolchain that works with all MCUXpresso SDKs. Easily import your SDK into the new toolchain to have 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 for easy cloning of existing SDK examples and demos, allowing users to easily leverage the existing software examples provided by the SDK for their own projects.

NOTE

In order to maintain compatibility with legacy FSL 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.

3 Development tools

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

- IAR Embedded Workbench for Arm version 8.22.2
- MDK-Arm Microcontroller Development Kit (Keil)® 5.24a
- Makefiles support with GCC revision 7-2017-q4-major from Arm Embedded
- MCUXpresso IDE v10.2.0

4 Supported development systems

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

Table 1. Supported MCU devices and development boards

Development boards	MCU devices
LPCXpresso802, LPCXpresso804, LPCXpresso812MAX, LPCXpresso824MAX, LPCXpresso845MAX	**LPC802M011JDH20**, LPC802M001JDH16, LPC802M001JDH20, LPC802M001JHI33, **LPC804M101JDH24**, LPC804M101JDH20, LPC804M111JDH24, LPC804M101JHI33, **LPC812M101JDH20**, LPC810M021FN8, LPC811M001JDH16, LPC812M101JDH16, LPC812M101JD20, LPC812M101JTB16, **LPC824M201JHI33**, LPC834M101FHI33, LPC832M101FDH20, LPC824M201JDH20, **LPC845M301JBD64**, LPC845M301JBD48, LPC845M301JHI48, LPC845M301JHI33

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
Documentation	<install_dir>/docs
Middleware	<install_dir>/middleware
DMA manager	<install_dir>/middleware/dma_manager
SDMMC card driver	<install_dir>/middleware/sdmmc
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
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
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, CMSIS, devices, documentation, middleware, and RTOS support.

6.1 Device support

The device folder contains all available software enablement for the specific System-on-Chip (SoC) subfamily. This folder includes clock-specific implementation, device register header file, device register feature header file, 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 simple debug console.

The device-specific header files provide a direct access to the MCU peripheral registers. The device header file provides an overall 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, 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.

The RTOS and middleware folders each contain examples demonstrating the use of the included source.

6.2 Middleware

6.2.1 CMSIS

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

7 MISRA compliance

All MCUXpresso SDK drivers comply to MISRA 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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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.
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.

Table continues on the next page...

Table 3. MISRA exceptions (continued)

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 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 17.7	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.

8.2 USBFS controller issue

Because of the USBFS controller design issues, the USB host suspend/resume demos (usb_suspend_resume_host_hid_mouse) of the full speed controller do not support the low speed device directly.

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.

9 Revision history

This table summarizes revisions to this document.

Table 4. Revision history

Revision number	Date	Substantive changes
0	05/2018	Initial release MCUXpresso SDK v.2.4.0
1	06/2018	Updated Section 4, "Supported development systems"

MCUXpresso SDK Release Notes Supporting LPCXpresso8XX

Change Logs

Contents

Driver Change Log 1

- USART . . . 1
- IOCON . . . 1
- CTIMER . . . 1
- CAPT . . . 1
- CRC . . . 1
- ADC . . . 2
- DAC . . . 2
- LPC_ACOMP . . . 2
- DMA . . . 2
- GPIO . . . 2
- PINT . . . 3
- SYSCON . . . 3
- I2C . . . 3
- MRT . . . 3
- INPUTMUX . . . 3
- SWM . . . 4
- SCTIMER . . . 4
- WKT . . . 4
- WWDT . . . 4
- SPI . . . 4

Contents

Title	Page Number
CLOCK	4
POWER	5
RESET	5

1 Driver Change Log

USART

The current USART driver version is 2.0.0.

- 2.0.0
  - Initial version.

IOCON

The current IOCON driver version is 2.0.0.

- 2.0.0
  - Initial version.

CTIMER

The current CTimer driver version is 2.0.1.

- 2.0.1
  - API Interface Change Added CTIMER_SetupPwmPeriod and CTIMER_UpdatePwmPulsePeriod API. These two APIs can set up the right PWM with high resolution.
- 2.0.0
  - Initial version.

CAPT

The current CAPT driver version is 2.0.0.

- 2.0.0
  - Initial version.

CRC

The current CRC driver version is 2.0.1.

- 2.0.1
  - Bug fix:
    - DATA and DATALL macro definition moved from header file to source file.
- 2.0.0
  - Initial version.

ADC

The current ADC driver version is 2.2.0.

- 2.2.0
  - Updated "ADC_DoSelfCalibration" API and "adc_config_t" structure to match LPC845.
- 2.1.0
  - Renamed "ADC_EnableShresholdCompareInterrupt" to "ADC_EnableThresholdCompare-Interrupt".
- 2.0.0
  - Initial version.

DAC

The current DAC driver version is 2.0.0.

- 2.0.0
  - Initial version.

LPC_ACOMP

The current LPC_ACOMP driver version is 2.0.0.

- 2.0.0
  - Initial version.

DMA

The current DMA driver version is 2.0.0.

- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.1.1.

- 2.1.1:
  - API interface changes:
    - Refined naming of API while keep all original APIs, marking them as deprecated. Original API will be removed in next release. The mainin change is update API with prefix of _-PinXXX() and _PorortXXX
- 2.1.0
  - Added GPIO initialize API.
- 2.0.0

  - Initial version.

PINT

The current PINT driver version is 2.0.1.

- 2.0.1
  - Bug fix:
    - Updated PINT driver to clear interrupt only in Edge sensitive.
- 2.0.0
  - Initial version.

SYSCON

The current SYSCON driver version is 2.0.0.

- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.1.

- 2.0.1
  - Improvements:
    - Added I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.0
  - Initial version.

MRT

The current MRT driver version is 2.0.0.

- 2.0.0
  - Initial version.

INPUTMUX

The current INPUTMUX driver version is 2.0.0.

- 2.0.0
  - Initial version.

SWM

The current SWM driver version is 2.0.0.

- 2.0.0
  - Initial version.
  - The API SWM_SetFixedMovablePinSelect() is targeted at the device that has PINASSIGNFIXED0 register, such as LPC804.

SCTIMER

The current SCTimer driver version is 2.0.0.

- 2.0.0
  - Initial version.

WKT

The current WKT driver version is 2.0.0.

- 2.0.0
  - Initial version.

WWDT

The current WWDT driver version is 2.0.0.

- 2.0.0
  - Initial version.

SPI

The current SPI driver version is 2.0.0.

- 2.0.0
  - Initial version.

CLOCK

The current CLOCK driver version is 2.0.2.

- 2.0.2
  - some minor fixes.
- 2.0.0

  - initial version.

POWER

The current POWER driver version is 2.0.0.

- 2.0.0
  - initial version.

RESET

The current RESET driver version is 2.0.0.

- 2.0.0
  - initial version.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number MCUXSDKLPC8XXRN
Revision 1, 06/2018