MCUXSDKMIMXRT117HRN

MCUXpresso SDK Release Notes for SLN-VIZN3D-IOT

Rev. 0 — 29 October 2021 Release Notes

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 LittleFS, other middleware packages. In addition to the base enablement, the MCUXpresso SDK is augmented with demo applications, driver example projects, and API documentation to help customers leverage the MCUXpresso SDK support.

For details on MCUXpresso SDK, see MCUXpresso-SDK: Software Development Kit for MCUXpresso.

2 MCUXpresso SDK

As part of the MCUXpresso software and tools, the MCUXpresso SDK is the evolution of the Kinetis SDK, including 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 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

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:

- Makefile support with GCC revision 10-2020-q4-major GCC10 from Arm Embedded
- MCUXpresso IDE v11.4.1

4 Supported development system

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

Table 1. Supported development system

Development boards	MCU devices
SLN-VIZN3D-IOT	**MIMXRT117HDVMAA**

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
Boot Applications	<install_dir>/boards/<board_name>/sln_boot_apps
cJSON	<install_dir>/middleware/cjson
CMSISArm Cortex®-M header files, DSP library source	<install_dir>/CMSIS
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Documentation	<install_dir>/docs
Driver, SoC header files, extension header files and feature header files, utilities	<install_dir>/devices/<device_name>
Smart-Lock Application	<install_dir>/boards/sln_vizn3d_iot/sln_vision_apps/smart_lock
LittleFS	<install_dir>/middleware/littlefs
lwIP Documentation	<install_dir>/docs/lwip
lwIP stack	<install_dir>/middleware/lwip
mbed TLS	<install_dir>/middleware/mbedtls
Peripheral Drivers	<install_dir>/devices/<device_name>/drivers
RTOS Kernel Code	<install_dir>/rtos
Solutions IoT Common Platform	<install_dir>/middleware/sln_iot_common_platform
Oasis	<install_dir>/middleware/oasis
Orbbec	<install_dir>/middleware/orbbec
Tools	<install_dir>/tools
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. The release package 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

For more information, see the *MCUXpresso SDK USB Stack User's Guide* (document MCUXSDKUSBSUG).

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.

6.2.6 Orbbec

The MCUXpresso SDK provides integration with the Orbbec U1S Astra SL1115L-M camera.

6.2.7 Oasis

The MCUXpresso SDK is shipped with the Oasis-Lite inference engine for secure face recognition.

7 MISRA compliance

All MCUXpresso SDK drivers comply to MISRA 2012 rules with exceptions in Table 3.

Table 3. MISRA exception rules

Exception rules	Description
Directive 4.4	Sections of code should not be commented out.

Table continues on the next page...

Table 3. MISRA exception rules (continued)

Exception rules	Description
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.

Table continues on the next page...

Table 3. MISRA exception rules (continued)

Exception rules	Description
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 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 <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 required by the user.

NOTE

`xxx` means core variants, such as, `cm0plus`, `cm33`, `cm4`, `cm33_nodsp`.

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.

8.3 RAM targets build issue in CMSIS BSP pack

The pack does not support different macro definitions for different targets. Therefore, all RAM targets for projects inside CMSIS BSP PACKs for RT10XX boards gets 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.

9 Known SDK issues

Table 4 lists the known issues in v1.0.0 of SLN-VIZN3D-IOT SDK.

Table 4. Known SDK issues

Issue #	Description	Impact	Workaround
RTVZN-795	Excessive “Look at camera” prompts during registration while in headless mode. This happens when the algorithm wrongly classifying the face as “side face.”	In headless mode, some user faces may incorrectly get classified as “side face” during registration despite looking straight at the camera. This makes it harder for those users to register their face. In the worst case, the registration operation will time out.	Try slowly moving the face in different directions while in front of the camera.
RTVZN-783	Fake face prompt during registration, in normal usage.	Some users may be presented with “fake face” prompts during registration/ recognition. This makes it harder for those users to register their face. In the worst case, the registration operation times out	Try moving slowly the face in any direction, in front of the camera.
RTVZN-778	Each press of the button queues sound.	Every time the SW button is pressed, the sound feedback is placed in a queue and	Not visible under normal usage

Table continues on the next page...

Table 4. Known SDK issues (continued)

Issue #	Description	Impact	Workaround
		eventually played back. If many buttons are pressed in quick succession, the audio is played much later.	
RTVZN-686	The board does not recognize some remotely registered users.	The board under certain lighting conditions does not recognize some users who registered themselves using the remote application.	None
RTVZN-660	"Invalid bluetooth packet" toast notification in the apk.	Remote application may display an “Invalid Bluetooth packet” notification during usage. This notification means that the desired action was not executed and must be retried.	Tap again the desired button.
RTVZN-631	User can get registered both remotely and locally.	Some users may be able to register themselves both using the remote application and on the board itself, resulting in a duplicate face listing for the same user.	None

10 Revision history

Table 5 table summarizes revisions to this document.

Table 5. Revision history

Revision number	Date	Substantive changes
0	29 October 2021	Initial release.

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: 29 October 2021

Document identifier: MCUXSDKMIMXRT117HRN

arm