NXP Semiconductors
User's Guide

Document Number: MCUXSDKLPC54114RN
Rev. 0, December 2018

MCUXpresso SDK Release Notes Supporting LPCXpresso54114

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 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.

Chapter 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, an 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.

Chapter 3
Development tools

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

- IAR Embedded Workbench for Arm version 8.32.1
- MDK-Arm Microcontroller Development Kit (Keil)® 5.26
- Makefiles support with GCC revision 7-2018-q2-update from Arm Embedded
- MCUXpresso IDE v10.3.0

Chapter 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
LPCXpresso54114	**LPC54114J256BD64,** LPC54114J256UK49, LPC54113J128BD64, LPC54113J256BD64, LPC54113J256UK49

Chapter 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
TinyCBOR	<install_dir>/rtos/amazon-freertos/lib/third_party/tinycbor
Demo applications	<install_dir>/boards/<board_name>/demo_apps
USB demo applications	<install_dir>/boards/<board_name>/usb_examples
Driver examples	<install_dir>/boards/<board_name>/driver_examples
CMSIS driver examples	<install_dir>/boards/<board_name>/cmsis_driver_examples
<install_dir>/boards/<board_name>/aws_examples	
emWin examples	<install_dir>/boards/<board_name>/emwin_examples
RTOS examples	<install_dir>/boards/<board_name>/rtos_examples
Multicore examples	<install_dir>/boards/<board_name>/multiprocessor_examples
Documentation	<install_dir>/docs
USB Documentation	<install_dir>/docs/usb
Middleware	<install_dir>/middleware
eRPC	<install_dir>/middleware/multicore/erpc
RPMSG lite	<install_dir>/middleware/multicore/rpmsg_lite
Multicore manager (MCMGR)	<install_dir>/middleware/multicore/mcmgr
Multicore stack	<install_dir>/middleware/multicore
USB stack	<install_dir>/middleware/usb
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
CMSIS drivers	<install_dir>/devices/<device_name>/cmsis_drivers
Utilities such as debug console	<install_dir>/devices/<device_name>/utilities
RTOS Kernel Code	<install_dir>/rtos
Tools	<install_dir>/tools
segger_systemview	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_segger_sysview

Table continues on the next page...

Table 2. Release contents (continued)

percepio_snapshot	<install_dir>/boards/<board>/rtos_examples/visualization/freertos_percepio_snapshot
jquery_ui_css	<install_dir>/middleware/multicore/tools/pam-manager/styles/jquery-ui-1.11.4.css
icon_images_jqueryui	<install_dir>/middleware/multicore/tools/pam-manager/styles/images/*
jquery_ui	<install_dir>/middleware/multicore/tools/pam-manager/js/jquery-ui-1.11.4.js
jquery	<install_dir>/middleware/multicore/tools/pam-manager/js/jquery-1.11.3.min.js
FileSaver_JS	<install_dir>/middleware/multicore/tools/pam-manager/js/FileSaver.js
icon_images_icon_edit_png	<install_dir>/middleware/multicore/tools/pam-manager/images/icon_edit.png
icon_images_icon_delete_png	<install_dir>/middleware/multicore/tools/pam-manager/images/icon_delete.png
icon_images_icon_add_png	<install_dir>/middleware/multicore/tools/pam-manager/images/icon_add.png
cpp_option_parser	<install_dir>/middleware/multicore/erpc/erpcgen/src/options.cpp
cpp_template	<install_dir>/middleware/multicore/erpc/erpcgen/src/cpptemplate
the_bus_pirate	<install_dir>/middleware/multicore/erpc/erpc_c/port/erpc_serial.h, middleware/multicore/erpc/erpc_c/port/erpc_serial.cpp
bosch_sensor_drivers.lpcxpresso54114	<install_dir>/boards/lpcxpresso54114/multicore_examples/low_power/sensors
bosch_sensor_drivers.lpcxpresso54102	<install_dir>/boards/lpcxpresso54102/multicore_examples/low_power/sensors

Chapter 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 CMSIScompliant 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.

6.2 Middleware

6.2.1 USB Type-C PD stack

See the *MCUXpresso SDK USB Type-C PD Stack User's Guide* (document MCUXSDKUSBPDUG) for more information.

NOTE

The USB TYPE-C PD stack supports IAR only.

6.2.2 USB stack

See the *MCUXpresso SDK USB Stack User's Guide* (document MCUXSDKUSBSUG) for more information.

6.2.2.1 Peripheral devices tested with USB Host stack

This table 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 SAMSUNG USB3.0 16GB
USB card reader/adapter	SSK TF adapter Kawau Multi Card Reader Kawau TF adapter Kawau SDHC card

Table continues on the next page...

Table 3. Peripheral devices (continued)

USB Mouse	DELL MS111-P DELL M066U0A DELL MUAVDEL8 TARGUS AMU76AP DELL MD56U0 DELL MS111-T RAPOO M110
USB Keyboard	DELL SK8135 DELL SK8115

6.2.3 RTOS

The MCUXpresso SDK is integrated with FreeRTOS OS.

6.2.4 CMSIS

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

6.2.5 emWin

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

6.2.6 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 and USB stack comply to MISRA 2012 rules with the following exceptions.

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	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.
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.

Table continues on the next page...

Table 4. MISRA exceptions (continued)

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.
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.

Table continues on the next page...

Table 4. MISRA exceptions (continued)

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 17.1	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.

Chapter 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 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 Type-C PD stack

The USB Type-C PD stack in the MCUXpresso SDK supports IAR only. These limitations cannot be explicitly shown on the KEX webpage.

8.4 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).

MCUXpresso SDK Release Notes Supporting LPCXpresso54114

Change Logs

Contents

Contents

1 Driver Change Log

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.

CRC

The current CRC driver version is 2.0.1.

- 2.0.1
  - Fixed KPSDK-13362. MDK compiler issue when writing to WR_DATA with -O3 optimize for time.
- 2.0.0
  - Initial version.

CTIMER

The current CTimer driver version is 2.0.2.

- 2.0.2
  - Add new API "CTIMER_GetTimerCountValue" to get the current timer count value.
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
  - Add new feature macro to update the API of ctimer driver for lpc8n04.
- 2.0.1
  - API Interface Change Added CTIMER_SetupPwmPeriod and CTIMER_UpdatePwmPulse-Period API. These two APIs can set up the right PWM with high resolution.
- 2.0.0
  - Initial version.

DMA

The current DMA driver version is 2.0.1.

- 2.0.1

  - Add volatile for dma descriptor member xfercfg to avoid optimization.
- 2.0.0
  - Initial version.

DMIC

The current DMIC driver version is 2.0.1.

- 2.0.1
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

FLASHIAP

The current FLASHIAP driver version is 2.0.3.

- 2.0.3
  - The flashiap driver is marked as deprecated and it will be removed in next release. All its APIs are moved to iap driver. The names of flashiap's APIs are updated from FLASHIAP_XXX() to IAP_XXX().
- 2.0.2
  - Added the API for extended flash signature
- 2.0.1
  - Remove two incorrect commands.
- 2.0.0
  - Initial version.

FLEXCOMM

The current FLEXCOMM driver version is 2.0.0.

- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.3.

- 2.0.3
  - Unify component full name to FLEXCOMM I2C(DMA/FREERTOS) Driver
- 2.0.2
  - Improvements: In slave IRQ:

1. Changed slave receive process to first set the I2C_SLVCTL_SLVCONTINUE_MASK to ack the received data, then do data receive.
2. Improved slave transmit process to set the I2C_SLVCTL_SLVCONTINUE_MASK immediately after write the data.

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

I2S

The current I2S driver version is 2.0.2.

- 2.0.2
  - Add ENABLE_IRQ handle after register i2s interrupt handle
- 2.0.1
  - Unify component full name to FLEXCOMM I2S(DMA) Driver
- 2.0.0
  - Initial version.

SPI

The current SPI driver version is 2.0.2.

- 2.0.2
  - Unify component full name to FLEXCOMM SPI(DMA/FREERTOS) Driver
- 2.0.1
  - Changed the data buffer from uint32_t to uint8_t which matches the real applications for SPI DMA driver.
- Added dummy data setup API to allow users to configure the dummy data to be transferred.
  - Added new APIs for half-duplex transfer function, users can send and receive data by one API in polling/interrupt/DMA way, and users can choose either transmit first or receivefirst. Besides, the PCS pin can be configured as assert status in transimission (between transmit and receive) by setting the isPcsAssertInTransfer to true.
- 2.0.0
  - Initial version.

USART

- The current USART driver version is 2.0.3.
- 2.0.3

  - New feature:
    - Add new API to allow users enable the CTS which determines whether CTS is used for flow control.
- 2.0.2
  - Bug Fix:
    - Fix the bug of transfer abort APIs can not disable the interrupts, the FIFOINTENSET register should not be used to disable the insterrupts, using FIFOINTENCLR register instead.
- 2.0.1
  - Unify component full name to FLEXCOMM USART(DMA/FREERTOS) Driver
- 2.0.0
  - Initial version.

FMEAS

The current FMEAS driver version is 2.0.0.

- 2.0.0
  - Initial version ported from LPCOpen.

GINT

The current GINT driver version is 2.0.01.

- 2.0.1
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.1.2.

- 2.1.2:
  - Remove deprecated APIs.
- 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.

INPUTMUX

The current INPUTMUX 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.

MRT

The current MRT driver version is 2.0.1.

- 2.0.1
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

PINT

The current PINT driver version is 2.0.3.

- 2.0.3 Add PINT_EnableCallbackByIndex/PINT_DisableCallbackByIndex APIs to enable/disable callback by index.
- 2.0.2
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
  - Bug fix:
    - Updated PINT driver to clear interrupt only in Edge sensitive.
- 2.0.0
  - Initial version.

RTC

The current RTC driver version is 2.0.0.

- 2.0.0
  - Initial version.

SCTIMER

The current SCTimer driver version is 2.0.1.

- 2.0.1
  - Add control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
  - Initial version.

WWDT

The current WWDT driver version is 2.1.0.

- 2.1.0
  - Add new parameter in configuration when initializing WWDT module, this patameter will allow user to deliver the WWDT clock freauency, and this parameter must be set.
- 2.0.0
  - Initial version.

UTICK

The current UTICK driver version is 2.0.1.

- 2.0.1
  - Add control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

SYSCON

The current SYSCON driver version is 2.0.0.

- 2.0.0
  - Initial version.

MAILBOX

The current MAILBOX driver version is 2.1.0.

- 2.1.0
  - Adds support for the LPC55S69 serie, cpu_id parameter can be newly assigned to kMAILBOX_CM33_Core0 or kMAILBOX_CM33_Core1.
- 2.0.0
  - Initial version.

CLOCK

The current CLOCK driver version is 2.0.4.

- 2.0.4
  - Bug Fix:
    - add missing main clock source for MCLK.
- 2.0.3
  - New Feature:
    - add get actual clock attach id api to allow users to obtain the actual clock source in target register.
  - Bug Fix:
    - The attach clock and get actual clock attach id apis should check combination of two clock source.
  - Optimization:
    - Make the judgement statments more clear.
    - Strengthen the compatibility of clock attatch id.
    - Remove some unmeaningful definitions and add some useful ones to enhance readability.
- 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.

COMMON

The current COMMON driver version is 2.0.1.

- 2.0.1
  - Remove the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
  - Add new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTION" for specific SoC which has no noncacheable sections, this will help avoid unnecessary

complex in link file and startup file.
  - Update the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
  - Initial version.

2 Middleware Change Log

FatFs for MCUXpresso SDK

The current version is FatFs R0.13a_rev0.

- R0.13a_rev0
  - Upgraded to version 0.13a. Added patch ff_13a_p1.diff.
- R0.12c_rev1
  - Add NAND disk support.
- R0.12c_rev0
  - Upgraded to version 0.12c and applied patches ff_12c_p1.diff and ff_12c_p2.diff.
- R0.12b_rev0
  - Upgraded to version 0.12b.
- R0.11a
  - Added glue functions for low-level drivers (SDHC, SDSPI, RAM, MMC). Modified diskio.c.
  - Added RTOS wrappers to make FatFs thread safe. Modified syscall.c.
  - Renamed ffconf.h to ffconf_template.h. Each application should contain its own ffconf.h.
  - Included ffconf.h into diskio.c to enable the selection of physical disk from ffconf.h by macro definition.
  - Conditional compilation of physical disk interfaces in diskio.c.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.1.0.

- 2.1.0
  - New features:
    - add host rndis support. example: lwip_dhcp_usb
    - enable usb3.0 support on device stack.
    - pd feature Add OM13790HOST support; Add auto policy feature; Print e-marked cable information;
- 2.0.1
  - Bug fix:
    - fixed some USB issues: fix msc cv test failed in msc examples.
    - Change the audio codec interfaces.
- 2.0.0
  - New features:
    - PTN5110N support.
  - Bug fix:
    - Added some comments, fixed some minor USB issues.
- 1.9.0
  - New features:
    - Examples:

            - usb_pd_alt_mode_dp_host
    - 1.8.2
        - Updated license.
    - 1.8.1
        - Bug fix:
            - Verified some hardware issues, support aruba_flashless.
    - 1.8.0
        - New features:
            - Examples:
                - usb_device_composite_cdc_vcom_cdc_vcom
                - usb_device_composite_hid_audio_unified
                - usb_pd_sink_battery
                - Changed usb_pd_battery to usb_pd_charger_battery.

```
- Bug fix:
  - Code clean up, removed some irrelevant code.
```

    - 1.7.0
        - New features:
            - USB PD stack support.
        - Examples
            - usb_pd
            - usb_pd_battery
            - usb_pd_source_charger
    - 1.6.3
        - Bug fix: -IP3511_HS driver control transfer sequence issue, enabled 3511 ip cv test.
    - 1.6.2
        - New features:
            - Multi instance support.
    - 1.6.1
        - New features:
        - Changed the struct variable address method for device_video_virtual_camera and host_phdc-_manager.
    - 1.6.0
        - New features:
            - Supported Device Charger Detect feature on usb_device_hid_mouse.
    - 1.5.0
        - New features:
            - Supported controllers
                - OHCI (Full Speed, Host mode)
                - IP3516 (High Speed, Host mode)
                - IP3511 (High Speed, Device mode)
            - Examples:
                - usb_lpm_device_hid_mouse
                - usb_lpm_device_hid_mouse_lite

        - usb_lpm_host_hid_mouse
- 1.4.0
  - New features:
    - Examples:
      - usb_device_hid_mouse/freertos_static
      - usb_suspend_resume_device_hid_mouse_lite
- 1.3.0
  - New features:
    - Supported roles
      - OTG
    - Supported classes
      - CDC RNDIS
    - Examples
      - usb_otg_hid_mouse
      - usb_device_cdc_vnic
      - usb_suspend_resume_device_hid_mouse
      - usb_suspend_resume_host_hid_mouse
- 1.2.0
  - New features:
    - Supported controllers
      - LPC IP3511 (Full Speed, Device mode)
- 1.1.0
  - Bug fix:
    - Fixed some issues in USB certification.
    - Changed VID and Manufacturer string to NXP.
  - New features:
    - Supported classes
      - Pinter
    - Examples:
      - usb_device_composite_cdc_msc_sdcard
      - usb_device_printer_virtual_plain_text
      - usb_host_printer_plain_text
- 1.0.1
  - Bug fix:
    - Improved the efficiency of device audio speaker by changing the transfer mode from interrupt to DMA, thus providing the ability to eliminate the periodic noise.
- 1.0.0
  - New features:
    - Supported roles
      - Device
      - Host
    - Supported controllers:
      - KHCI (Full Speed)
      - EHCI (High Speed)
    - Supported classes:

    - AUDIO
    - CCID
    - CDC
    - HID
    - MSC
    - PHDC
    - VIDEO
  - Examples:
    - usb_device_audio_generator
    - usb_device_audio_speaker
    - usb_device_ccid_smart_card
    - usb_device_cdc_vcom
    - usb_device_cdc_vnic
    - usb_device_composite_cdc_msc
    - usb_device_composite_hid_audio
    - usb_device_composite_hid_mouse_hid_keyboard
    - usb_device_hid_generic
    - usb_device_hid_mouse
    - usb_device_msc_ramdisk
    - usb_device_msc_sdcard
    - usb_device_phdc_weighscale
    - usb_device_video_flexio_ov7670
    - usb_device_video_virtual_camera
    - usb_host_audio_speaker
    - usb_host_cdc
    - usb_host_hid_generic
    - usb_host_hid_mouse
    - usb_host_hid_mouse_keyboard
    - usb_host_msd_command
    - usb_host_msd_fatfs
    - usb_host_phdc_manager
    - usb_keyboard2mouse
    - usb_pin_detect_hid_mouse

3 RTOS Change Log

FreeRTOS for MCUXpresso SDK

The current version is FreeRTOS 9.0.0. Original package is available at freertos.org.

- 9.0.0_rev3
  - New features:
    - Tickless idle mode support for Cortex-A7. Add fsl_tickless_epit.c and fsl_tickless_generic.h in portable/IAR/ARM_CA9 folder.
    - Enabled float context saving in IAR for Cortex-A7. Added configUSE_TASK_FPU_SUPPORT macros. Modified port.c and portmacro.h in portable/IAR/ARM_CA9 folder.
  - Other changes:
    - Transformed ARM_CM core specific tickless low power support into generic form under freertos/Source/portable/low_power_tickless/.
- 9.0.0_rev2
  - New features:
    - Enabled MCUXpresso thread aware debugging. Add freertos_tasks_c_additions.h and configINCLUDE_FREERTOS_TASK_C_ADDITIONS_H and configFRTOS_MEMORY_SCHEME macros.
- 9.0.0_rev1
  - New features:
    - Enabled -flto optimization in GCC by adding **attribute**((used)) for vTaskSwitchContext.
    - Enabled KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD_STACK_HIGH_ADDRESS macro. Modified files are task.c and FreeRTOS.h.
- 9.0.0_rev0
  - New features:
    - Example freertos_sem_static.
    - Static allocation support RTOS driver wrappers.
  - Other changes:
    - Tickless idle rework. Support for different timers is in separated files (fsl_tickless_systick.c, fsl_tickless_lptmr.c).
    - Removed configuration option configSYSTICK_USE_LOW_POWER_TIMER. Low power timer is now selected by linking of apropriate file fsl_tickless_lptmr.c.
    - Removed configOVERRIDE_DEFAULT_TICK_CONFIGURATION in RVDS port. Use of **attribute**((weak)) is the preferred solution. Not same as _weak!
- 8.2.3
  - New features:
    - Tickless idle mode support.
    - Added template application for Kinetis Expert (KEx) tool (template_application).
  - Other changes:
    - Folder structure reduction. Keep only Kinetis related parts.

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.

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

Date of release: December 2018
Document identifier: MCUXSDKLPC54114RN

arm