
MCUXpresso SDK Release Notes Supporting Ipcxpresso804

Change Logs

NXP Semiconductors



Contents

Driver Change Log

CLOCK	1
POWER	1
RESET	2
COMMON	2
SPI	4
USART	5
SWM	6
SYSCON	7

Middleware Change Log

FreeMASTER Communication Driver	8
SAFETY_IEC60730B for KSDK	9

1 Driver Change Log

CLOCK

The current CLOCK driver version is 2.3.3.

- 2.3.3
 - Improvements
 - * Added lost comments for some enumerations.
- 2.3.2
 - Improvements
 - * Used "offsetof" macro to get the offset of the structure element from the beginning of the structure.
- 2.3.1
 - Bug Fixes
 - * Fixed MISRA C-2012 rule 10.1, rule 10.4, rule 10.8, rule 16.4 and so on.
- 2.3.0
 - New feature:
 - * Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.2.0
 - Replace the delay function
- 2.1.0
 - New feature
 - * Adding new API CLOCK_DelayAtLeastUs() to implement a delay function which allows users to set delay in unit of microsecond.
- 2.0.3
 - add api to get uart clock frequency.
 - add api to set fractional multiplier value.
- 2.0.2
 - some minor fixes.
- 2.0.0
 - initial version.

POWER

The current POWER driver version is 2.1.0.

- 2.1.0
 - New features
 - * Added BOD control APIs.
- 2.0.4
 - Bug Fixes
 - * Fixed the typo "Enbale", correcting it as "Enable".
- 2.0.3

- Bug Fixes
 - * Fixed doxygen warnings(remove wrong character in annotation).
- 2.0.2
 - New Features
 - * Added the Enable/DisableDeepSleepIRQ() to enable/disable pin wake up.
- 2.0.1
 - Improvements
 - * Updated power drive to support PMU.
- 2.0.0
 - initial version.

RESET

The current RESET driver version is 2.0.1.

- 2.0.1
 - Update component full_name to "Reset Driver".
- 2.0.0
 - initial version.

COMMON

The current COMMON driver version is 2.4.0.

- 2.4.0
 - New Features
 - * Added EnableIRQWithPriority, IRQ_SetPriority, and IRQ_ClearPendingIRQ for ARM.
 - * Added MSDK_EnableCpuCycleCounter, MSDK_GetCpuCycleCount for ARM.
- 2.3.3
 - New Features
 - * Added NETC into status group.
- 2.3.2
 - Improvements
 - * Make driver aarch64 compatible
- 2.3.1
 - Bug Fixes
 - * Fixed MAKE_VERSION overflow on 16-bit platforms.
- 2.3.0
 - Improvements
 - * Split the driver to common part and CPU architecture related part.
- 2.2.10
 - Bug Fixes
 - * Fixed the ATOMIC macros build error in cpp files.
- 2.2.9

- Bug Fixes
 - * Fixed MISRA C-2012 issue, 5.6, 5.8, 8.4, 8.5, 8.6, 10.1, 10.4, 17.7, 21.3.
 - * Fixed SDK_Malloc issue that not allocate memory with required size.
- 2.2.8
 - Improvements
 - * Included stddef.h header file for MDK tool chain.
 - New Features:
 - * Added atomic modification macros.
- 2.2.7
 - Other Change
 - * Added MECC status group definition.
- 2.2.6
 - Other Change
 - * Added more status group definition.
 - Bug Fixes
 - * Undef __VECTOR_TABLE to avoid duplicate definition in cmsis_clang.h
- 2.2.5
 - Bug Fixes
 - * Fixed MISRA C-2012 rule-15.5.
- 2.2.4
 - Bug Fixes
 - * Fixed MISRA C-2012 rule-10.4.
- 2.2.3
 - New Features
 - * Provided better accuracy of SDK_DelayAtLeastUs with DWT, use macro SDK_DELAY_USE_DWT to enable this feature.
 - * Modified the Cortex-M7 delay count divisor based on latest tests on RT series boards, this setting lets result be closer to actual delay time.
- 2.2.2
 - New Features
 - * Added include RTE_Components.h for CMSIS pack RTE.
- 2.2.1
 - Bug Fixes
 - * Fixed violation of MISRA C-2012 Rule 3.1, 10.1, 10.3, 10.4, 11.6, 11.9.
- 2.2.0
 - New Features
 - * Moved SDK_DelayAtLeastUs function from clock driver to common driver.
- 2.1.4
 - New Features
 - * Added OTFAD into status group.
- 2.1.3
 - Bug Fixes
 - * MISRA C-2012 issue fixed.
 - Fixed the rule: rule-10.3.
- 2.1.2

- Improvements
 - * Add SUPPRESS_FALL_THROUGH_WARNING() macro for the usage of suppressing fallthrough warning.
- 2.1.1
 - Bug Fixes
 - * Deleted and optimized repeated macro.
- 2.1.0
 - New Features
 - * Added IRQ operation for XCC toolchain.
 - * Added group IDs for newly supported drivers.
- 2.0.2
 - Bug Fixes
 - * MISRA C-2012 issue fixed.
 - Fixed the rule: rule-10.4.
- 2.0.1
 - Improvements
 - * Removed the implementation of LPC8XX Enable/DisableDeepSleepIRQ() function.
 - * Added new feature macro switch "FSL_FEATURE_HAS_NO_NONCACHEABLE_SECTION" for specific SoCs which have no noncacheable sections, that helps avoid an unnecessary complex in link file and the startup file.
 - * Updated the align(x) to **attribute**(aligned(x)) to support MDK v6 armclang compiler.
- 2.0.0
 - Initial version.

SPI

The current SPI driver version is 2.0.5.

- 2.0.5
 - Bug Fixes
 - * Fixed bug that the transfer configuration does not take effect after the first transfer.
- 2.0.4
 - Bug Fixes
 - * Fixed the issue that when transfer finish callback is invoked TX data is not sent to bus yet.
- 2.0.3
 - Improvements
 - * Added timeout mechanism when waiting certain states in transfer driver.
 - * Fixed MISRA 10.4 issue.
- 2.0.2
 - Bug Fixes
 - * Fixed Coverity issue of incrementing null pointer in SPI_MasterTransferNonBlocking.
 - * Fixed MISRA issues.
 - Fixed rules 10.1, 10.3, 10.4, 10.6, 14.4.
 - New Features

- * Added enumeration for dataWidth.
- 2.0.1
 - Bug Fixes
 - * Added wait mechanism in SPI_MasterTransferBlocking() API, which checks if master SPI becomes IDLE when the EOT bit is set before returning. This confirms that all data will be sent out by SPI master.
 - * Fixed the bug that the EOT bit couldn't be set when only one frame was sent in polling mode and interrupt transfer mode.
 - New Features
 - * Added macro gate "FSL_SDK_ENABLE_SPI_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs, which helps reduce the code size when no nonblocking transfer is used. Enabled default configuration.
 - * Added a control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

USART

The current USART driver version is 2.5.0.

- 2.5.0
 - New Features
 - * Supported new feature of rx idle timeout.
- 2.4.0
 - Improvements
 - * Used separate data for TX and RX in usart_transfer_t.
 - Bug Fixes
 - * Fixed bug that when ring buffer is used, if some data is received in ring buffer first before calling USART_TransferReceiveNonBlocking, the received data count returned by USART_TransferGetReceiveCount is wrong.
- 2.3.0
 - New Features
 - * Modified usart_config_t, USART_Init and USART_GetDefaultConfig APIs so that the hardware flow control can be enabled during module initialization.
- 2.2.0
 - Improvements
 - * Added timeout mechanism when waiting for certain states in transfer driver.
 - * Fixed MISRA 10.4 issues.
- 2.1.1
 - Bug Fixes
 - * Fixed the bug that in USART_SetBaudRate best_diff rather than diff should be used to compare with calculated baudrate.
 - * Eliminated IAR pa082 warnings from USART_TransferGetRxRingBufferLength and USART_TransferHandleIRQ.

- * Fixed MISRA issues.
- Improvements
 - * Rounded up the calculated sbr value in USART_SetBaudRate to achieve more accurate baudrate setting.
 - * Modified USART_ReadBlocking so that if more than one receiver errors occur, all status flags will be cleared and the most severe error status will be returned.
- 2.1.0
 - New Features
 - * Added new APIs to allow users to configure the USART continuous SCLK feature in synchronous mode transfer.
- 2.0.1
 - Bug Fixes
 - * Fixed the repeated reading issue of the STAT register while dealing with the IRQ routine.
 - New Features
 - * Added macro gate "FSL_SDK_ENABLE_USART_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs, which helps reduce the code size when no nonblocking transfer is used. Enabled default configuration.
 - * Added a control macro to enable/disable the RESET and CLOCK code in current driver.
 - * Added macro switch gate "FSL_SDK_USART_DRIVER_ENABLE_BAUDRATE_AUTO_GENERATE" to enable/disable the baud rate to generate automatically. Disabling this feature will help reduce the code size to a certain degree. Default configuration enables auto generating of baud rate.
 - * Added the check of baud rate while initializing the USART. If the baud rate calculated is not precise, the software assertion will be triggered.
 - * Added a new API to allow users to enable the CTS, which determines whether CTS is used for flow control.
- 2.0.0
 - Initial version.

SWM

The current SWM driver version is 2.1.1.

- 2.1.1
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 10.1 and 10.3.
- 2.1.0
 - New Features
 - * Supported Flextimer function pin assign.
- 2.0.2
 - Bug Fixes
 - * MISRA C-2012 issue fixed: rule 14.3.
- 2.0.1
 - Bug Fixes

- * MISRA C-2012 issue fixed: rule 10.1, 10.3, and 10.4.
- 2.0.0
 - Initial version.
 - The API SWM_SetFixedMovablePinSelect() is targeted at the device that has PINASSIGNFIXED0 register, such as LPC804.

SYSCON

The current SYSCON driver version is 2.0.1.

- 2.0.1
 - Bug Fixes
 - * Fixed issue for MISRA-2012 check.
 - Fixed rule 10.4.
- 2.0.0
 - Initial version.

2 Middleware Change Log

FreeMASTER Communication Driver

Current version is 3.0.6. Visit <https://www.nxp.com/freemaster> for more information. Reach out for a support at <https://community.nxp.com/community/freemaster>.

- 3.0.0
 - Initial version of FreeMASTER driver reworked from a standalone package to MCUXpresso SDK middleware.
 - This driver version supports new version V4 of FreeMASTER serial communication protocol.
 - Supports UART, LPUART, USART, MINIUSART, FlexCAN, USB-CDC and JTAG/BDM communication.
 - Initial version was tested with the following boards: evkmimxrt1060, frdmk64f, frdmke15z, frdmkl28z, lpcxpresso54628 lpcxpresso55s69, lpcxpresso845max and twrk64f120m.
 - Use with FreeMASTER PC Host tool version 2.5 or later.
- 3.0.1
 - FreeMASTER driver extended to support wide range of Kinetis, LPC and i.MX-RT platforms.
 - Low-level communication drivers also available for few non-SDK NXP platforms like S12Z, S32x and more.
 - Use with FreeMASTER PC Host tool version 3.0 or later.
- 3.0.2
 - FreeMASTER driver support of DSC56F800EX and S12 platforms extended.
 - Removed dependency on C99 compiler features.
 - Use with FreeMASTER PC Host tool version 3.0.2 or later.
- 3.0.3
 - General update for SDK 2.9.0
 - fmstr_any demo added to selected platforms - use with MCUXpresso SDK and FreeMASTER peripheral configuration tool.
 - New example.pmp project file embedded into application flash storage.
 - USB-CDC implementation fixed, new JTAG EOnCE communication interface added to DSC 56F800E family.
 - Use with FreeMASTER PC Host tool version 3.0.3 or later. Version 3.1.x is recommended.
- 3.0.4
 - Fixed component dependency logic of FreeMASTER driver.
 - Use with FreeMASTER PC Host tool version 3.1.x
- 3.0.5
 - General update for SDK 2.11 and 2.12
 - New TCP and UDP support with lwIP stack
 - New communication over Segger RTT interface
 - Add fmstr_net and fmstr_wifi examples for selected i.MX-RT platforms
 - Add fmstr_rtt example for selected platforms
 - Fixed negative recorder threshold trigger processing
- 3.0.6

- General update for SDK 2.13
- Use of new Ethernet MDIO driver concept.
- Support of ENET and NETC Ethernet modules in the fmstr_net example application.

SAFETY_IEC60730B for KSDK

Current version is 1.1.0

- 1.1.0
 - Initial version.

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 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, Freescale, the Freescale logo, Kinetis, Processor Expert, and Tower are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, Cortex, Keil, Mbed, Mbed Enabled, and Vision 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.

© 2021 NXP B.V.

