
MCUXpresso SDK Release Notes Supporting Ipcxpresso55s69

Change Logs

NXP Semiconductors



Contents

Driver Change Log

CLOCK	1
RESET	1
RESET	1
EDMA_SOC	1
COMMON	1
LPADC	3
PRINCE	5
RNG	5
SDIF	6

Middleware Change Log

emWin library	9
FatFs for MCUXpresso SDK	9
FreeMASTER Communication Driver	10
lwIP for MCUXpresso SDK	11
mbedTLS for MCUXpresso SDK	15
MOTOR_CONTROL for KSDK	15
Multicore SDK	16
NTAG I2C plus library	24
RTCESL for KSDK	24
SAFETY_IEC60730B for KSDK	24

Title	Page No.
Host SDIF driver for MCUXpresso SDK	24
MMC Card driver for MCUXpresso SDK	25
SD Card driver for MCUXpresso SDK	28
SDIO Card driver for MCUXpresso SDK	30
Secure Element hostlib for KSDK	32
Trusted Firmware M (TF-M) for MCUXpresso SDK	34
USB stack for MCUXpresso SDK	34

Component Change Log

CODEC	41
WM8904	42
SGTL5000	44
DA7212	45
CS42888	46
SERIAL_MANAGER	46

1 Driver Change Log

CLOCK

The current CLOCK driver version is 1.0.0.

- 1.0.0
 - initial version.

RESET

The current RESET driver version is 1.0.0.

- 1.0.0
 - Initial version.

RESET

The current RESET driver version is 1.0.0.

- 1.0.0
 - Initial version.

EDMA_SOC

The current EDMA_SOC driver version is 1.0.0.

- 1.0.0
 - 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.

LPADC

The current LPADC driver version is 2.6.0.

- 2.6.0
 - Improvements
 - * Added the API LPADC_SetOffset12BitValue() to configure 12bit ADC conversion offset trim value manually.
 - * Added the API LPADC_SetOffset16BitValue() to configure 16bit ADC conversion offset

- trim value manually.
 - * Added API to set offset calibration mode.
 - * Added configuration of alternate channel.
 - * Updated auto calibration API and added calibration value conversion API.
- 2.6.0
 - New feature
 - * Added API LPADC_EnableHardwareTriggerCommandSelection() to enable trigger commands controlled by ADC_ETC.
- 2.5.1
 - Bug Fixes
 - * Fixed some typos in Lpadc driver comments.
- 2.5.0
 - Improvements
 - * Added missing items to enable trigger interrupts.
- 2.4.0
 - New features
 - * Added APIs to get/clear trigger status flags.
- 2.3.0
 - Improvements
 - * Removed LPADC_MeasureTemperature() function for the LPADC supports different temperature sensor calculation equations.
- 2.2.1
 - Improvements
 - * Optimized LPADC_MeasureTemperature() function to support the specific series with flash solidified calibration value.
 - * Clean doxygen warnings.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.3, rule 10.8 and rule 17.7.
- 2.2.0
 - New Feature
 - * Added API LPADC_MeasureTemperature() to get correct temperature from the internal sensor.
 - Improvements
 - * Separated lpadc_conversion_resolution_mode_t with related feature macro.
 - Bug Fixes
 - * Fixed the violations of MISRA C-2012 rules:
 - Rule 10.3, 10.4, 10.6, 10.7 and 17.7.
- 2.1.1
 - Improvements
 - * Updated the gain calibration formula.
 - * Used feature to segregate the new item kLPADC_TriggerPriorityPreemptSubsequently.
- 2.1.0
 - New Features
 - * Added the API LPADC_SetOffsetValue() to support configure offset trim value manually.
 - * Added the API LPADC_DoOffsetCalibration() to do offset calibration independently.

- Improvements
 - * Improved the usage of macros and removed invalid macros.
- 2.0.2
 - Improvements
 - * Added support for platforms with 2 FIFOs and different calibration measures.
- 2.0.1
 - Bug Fixes
 - * Ensured the API LPADC_SetConvCommandConfig configure related registers correctly.
- 2.0.0
 - Initial version.

PRINCE

The current PRINCE driver version is 2.3.2.

- Version 2.3.2
 - Fix documentation of enumeration.
 - Extend PRINCE example.
- Version 2.3.1
 - Fix MISRA-2012 issues.
 - Add support for LPC55S0x series
- Version 2.3.0
 - Add support for LPC55S1x and LPC55S2x series
- Version 2.2.0
 - Add runtime checking of the A0 and A1 rev. of LPC55Sxx serie to support both silicone revisions.
- Version 2.1.0
 - Update for the A1 rev. of LPC55Sxx serie.
- 2.0.0
 - Initial version.

RNG

The current RNG driver version is 2.0.3.

- 2.0.3
 - Modified RNG_Init and RNG_GetRandomData functions, added rng_accumulateEntropy and rng_readEntropy functions. These changes are reflecting recommended usage of RNG according to device UM
- 2.0.2
 - Add RESET_PeripheralReset function inside RNG_Init and RNG_Deinit functions.
- 2.0.1
 - Fix MISRA C-2012 issue.
- 2.0.0

- Initial version.

SDIF

The current SDIF driver version is 2.1.0

- 2.1.0
 - Improvements
 - * Removed redundant member endianMode in sdif_config_t.
 - * Added error status check in function SDIF_WaitCommandDone.
 - * Fixed the read fifo data incomplete issue in interrupt non-dma mode.
- 2.0.15
 - Bug Fixes
 - * Cleared the interrupt status before enable the interrupt to avoid interrupt generate unexpectedly.
 - * Fixed the SDIF_ReadDataPortBlocking blocking at wrong condition issue.
 - Improvements
 - * Enabled the functionality of timeout parameter in SDIF_SendCommand.
 - * Added the error recovery while sending sync clock command timeout.
- 2.0.14
 - Improvements
 - * Used different status code for command and data interrupt callback.
 - Bug Fixes
 - * Fixed the DMA descriptor attribute field unset when configuring the current transfer DMA descriptor issue which may cause the transfer terminate unexpected.
- 2.0.13
 - Improvements
 - * Disabled redundant interrupt per different transfer request.
 - * Disabled interrupt and reset command/data pointer in handle when transfer completes.
 - Bug Fixes
 - * Fixed the PA082 build warning.
 - * Fixed violations of the MISRA C-2012 rules 14.4, 17.7, 10.4, 10.3, 10.8, 14.3, 10.1, 16.4, 15.7, 12.2, 11.3, 11.9.
- 2.0.12
 - Bug Fixes
 - * Fixed the issue that SDIF_ConfigClockDelay didn't reset the delay field before write.
 - * Removed useless fifo reset code in transfer function.
 - * Fixed the divider overflow issue in function SDIF_SetCardClock.
- 2.0.11
 - Improvements
 - * Added API SDIF_GetEnabledInterruptStatus/SDIF_GetEnabledDMAInterruptStatus and used in SDIF_TransferHandleIRQ.
 - * Removed useless members interruptFlags/dmaInterruptFlags in the sdif_handle_t.
 - * Improved SDIF_SendCommand with return success directly when timeout is 0.

- * Added timeout error check when sending update clock command in SDIF_SetCardClock.
 - * Removed START_CMD status polling for normal command sending in SDIF_Transfer-Blocking/SDIF_TransferNonBlocking.
 - * Disabled timeout parameter in function SDIF_SendCommand.
- Bug Fixes
 - * Added delay cycle for the default speed mode(400 K and 25 M) to fix the timing issue when different AHB clocks are configured.
- 2.0.10
 - Bug Fixes
 - * Fixed the issue that API SDIF_EnableCardClock could not clear the clock enable bit.
- 2.0.9
 - Bug Fixes
 - * Fixed MDK 66-D warning.
- 2.0.8
 - New Features
 - * Added control macro to enable/disable the RESET and CLOCK code in current driver.
 - * Disabled useless interrupt while DMA is used.
 - * Updated SDIF driver for one instance support two cards.
- 2.0.7
 - Bug Fixes
 - * Enlarged the timeout value to avoid a command conflict issue.
- 2.0.6
 - Bug Fixes
 - * Removed assert(srcClock_Hz <= FSL_FEATURE_SDIF_MAX_SOURCE_CLOCK).
 - * Used hardware reset instead of software reset during initialization.
- 2.0.5
 - New Features
 - * Added non-word aligned data address and DMA descriptor address transfer support. Once one of the above addresses is not aligned, switch to host transfer mode.
 - Bug Fixes
 - * Fixed the issue that DMA suspended during initialization.
 - * Removed useless memset function call.
- 2.0.4
 - Improvements
 - * Added cardInserted/cardRemoved callback function.
 - * Added host base address/user data parameter for all call back functions.
- 2.0.3
 - Improvements
 - * Improved Clock Delay macro to allow the user to redefine and remove useless delay for clock below 25 MHz.
- 2.0.2
 - Bug Fixes
 - * Fixed the issue that the status flag could not be cleared entirely after transfer complete.
- 2.0.1
 - New Features

- * Improved interrupt transfer callback.
- Bug Fixes
 - * Added assert to limit the SDIF source clock below 52 MHz.
- 2.0.0
 - Initial version.

2 Middleware Change Log

emWin library

The currently supported version is 6.28

- v6.28
 - upgraded to v6.28
- v6.24_rev1
 - recompiled cm33 library with fpu single precision
 - added cm7_sp library for Cortex M7 with sp fpu for IAR
- v6.24
 - upgraded to v6.24
- v6.16c
 - upgraded to v6.16c
 - updated temperature_control demo generated by AppWizard
- v6.14d
 - upgraded to v6.14d
- v6.10f
 - upgraded to v6.10f

FatFs for MCUXpresso SDK

Current version is FatFs R0.14b_rev0.

- R0.14b_rev1
 - Applied patches from <http://elm-chan.org/fsw/ff/patches.html>
- R0.14b_rev0
 - Upgraded to version 0.14b
- R0.14a_rev0
 - Upgraded to version 0.14a
 - Applied patch ff14a_p1.diff and ff14a_p2.diff
- R0.14_rev0
 - Upgraded to version 0.14
 - Applied patch ff14_p1.diff and ff14_p2.diff
- R0.13c_rev0
 - Upgraded to version 0.13c
 - Applied patches ff_13c_p1.diff, ff_13c_p2.diff, ff_13c_p3.diff and ff_13c_p4.diff.
- R0.13b_rev0
 - Upgraded to version 0.13b
- 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.

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.

lwIP for MCUXpresso SDK

Lightweight IP (lwIP) is a small independent implementation of the TCP/IP protocol suite. Source code included in this SDK is based on development version 2.2.0.dev taken from 3rd party lwIP GIT repository. The webpage <https://git.savannah.nongnu.org/cgit/lwip.git> allows to browse the repository and also contains URLs for its cloning. The development versions (X.Y.Z.dev) do not refer to a single source code snapshots. To avoid ambiguity, change log below contains SHA-1 hashes of GIT commits used when importing the code into the SDK.

- 2.2.0_rev7
 - New features:
 - * Ported lwIP 2.2.0.dev (2022-05-09, branch: master, SHA-1: 239918ccc173cb2c2a62f41a40fd893f571) to MCUXpresso SDK.
 - * Added function ethernetif_probe_link() which reads actual link, speed and duplex settings from phy and passes them to driver. Stack could be set to call this function periodically by setting ETH_LINK_POLLING_INTERVAL_MS to value higher than zero.
 - * Added helper functions ethernetif_wait_linkup() and ethernetif_wait_ipv4_valid() to allow blocking of RTOS task or bare metal application until link is up or IPv4 address becomes valid.
 - * Added NETC adaptation layer.
 - * Processing of rx packets under RTOS moved from ISR to a separate task to improve system reaction times. Switch back to old behavior can be done by setting ETH_DO_RX_IN_SEPARATE_TASK macro to 0.
 - Bug fixes:
 - * port: Fixed copying of pbuf contents. Previous code was using an incorrect end condition and could result in the overrun of the destination buffer if more packets were on the queue.
 - * port: Delegating pbuf_free calls to tcpip_thread via pbuf_free_callback where possible (RTOS), ensured pbuf_free is not called from interrupt context when LWIP_ALLOW_MEM_FREE_FROM_OTHER_CONTEXT is not set (bare metal).
 - * port/enet_ethernetif_qos.c - Fixed ENET_RXBD_NUM which was used instead of ENE-

T_TXBD_NUM.

- * port/enet_ethernetif_qos.c - Fixed buffer alignment to be at least 64.
- * src/apps/lwiperf: Fixed IPv6 TCP TX throughput lower than IPv4 by modifying maximum segment size to avoid sending two segments instead of one.
- * src/apps/lwiperf: Out-of-order datagrams in UDP RX server mode are counted to the throughput.
- * src/apps/httpsrv: Implemented receive timeouts on sockets.
- * src/apps/httpsrv: Don't assert on HTTP session task creation failure.
- * src/apps/httpsrv: Fixed build with IPv6 enabled.
- * src/apps/httpsrv: Updated endianness macros required for websocket SHA generation.
- * src/apps/httpsrv: Added missing includes.
- 2.2.0_rev6
 - New features:
 - * Ported lwIP 2.2.0.dev (2022-03-25, branch: master, SHA-1: 124dc0a64ef5d7c14a27e3115e5888df65) to MCUXpresso SDK.
 - * Implemented leaving of multicast groups on ENET and ENET QOS.
- 2.2.0_rev5
 - New features:
 - * Ported lwIP 2.2.0.dev (2021-05-11, branch: master, SHA-1: 7ec4e9be304e7f8953740f10b2c810a292) to MCUXpresso SDK.
 - * LPC ENET adaptation layer allocates more buffers for frame reception now. Previously the number of receive buffers was determined by ENET_RXBD_NUM, which defaults to 5. It is determined by ENET_RXBUFF_NUM now, which is 2 * ENET_RXBD_NUM by default. Increase was needed because the actual version of LPC ENET driver always hold ENET_RXBD_NUM number of buffers and few additional buffers are needed for passing zero-copy frame data to lwIP. If this takes too much memory in your application, you can counteract by decreasing PBUF_POOL_SIZE, since PBUF_POOL is used only for transmission when LPC ENET, Kinetis ENET or ENET QOS is used.
- 2.2.0_rev4
 - New features:
 - * Ported lwIP 2.2.0.dev (2021-03-05, branch: master, SHA-1: 0056522cc974d2be2005c324f37187b5b3) to KSDK 2.0.0.
 - * LWIP_DHCP_DOES_ACD_CHECK option default changed to 0 (disabled):
 - Although the ACD check makes getting IP address from DHCP more robust, it added several seconds delay at startup of all applications which use DHCP.
 - This feature was not present in earlier versions of lwIP.
 - * ENET QOS adaptation layer - implemented zero-copy on receive.
 - * Kinetis ENET and ENET QOS adaptation layers allocate more buffers for frame reception now. Previously the number of receive buffers was determined by ENET_RXBD_NUM, which defaults to 5. It is determined by ENET_RXBUFF_NUM now, which is 2 * ENET_RXBD_NUM by default. Increase was needed because the actual version of Kinetis ENET and ENET QOS drivers always hold ENET_RXBD_NUM number of buffers and few additional buffers are needed for passing zero-copy frame data to lwIP. If this takes too much memory in your application, you can counteract by decreasing PBUF_POOL_SIZE, since PBUF_POOL is used only for transmission when Kinetis ENET or ENET

QOS is used.

- * Removed ethernetif_config_t.non_dma_memory field which was required to configure memory ranges unusable by ENET DMA on LPC devices. The setting has been replaced by BOARD_ENET_NON_DMA_MEMORY_ARRAY macro.
- 2.2.0_rev3
 - New features:
 - * Ported lwIP 2.2.0.dev (2020-07-07, branch: master, SHA-1: c385f31076b27efb8ee37f00cb5568783a) to KSDK 2.0.0.
- 2.2.0_rev2
 - New features:
 - * Kinetis ENET adaptation layer - implemented zero-copy on receive.
 - * lwiperf - counter of transferred bytes extended from 32 to 64 bit
 - Bug fixes:
 - * Fixed restarting Auto IP from DHCP.
- 2.2.0_rev1
 - New features:
 - * Ported lwIP 2.2.0.dev (2019-12-12, branch: master, SHA-1: 555812dcec38c9a2ef1ef9b31816291549) to KSDK 2.0.0.
 - * Implemented LWIP_ASSERT_CORE_LOCKED related functions in sys_arch.c. It can be enabled in lwipopts.h:


```

              • #define LWIP_ASSERT_CORE_LOCKED() sys_check_core_locking()
              • #define LWIP_MARK_TCPIP_THREAD() sys_mark_tcpip_thread()
                // if NO_SYS == 0
              • #define LOCK_TCPIP_CORE() sys_lock_tcpip_core() // if N-
                O_SYS == 0 and LWIP_TCPIP_CORE_LOCKING == 1
              • #define UNLOCK_TCPIP_CORE() sys_unlock_tcpip_core() //
                if NO_SYS == 0 and LWIP_TCPIP_CORE_LOCKING == 1
              
```
- 2.1.2_rev5
 - New features:
 - * Implemented TCP_USER_TIMEOUT socket option.
 - * Implemented SIOCOUTQ ioctl.
- 2.1.2_rev4
 - New features:
 - * Ported lwIP 2.1.3.dev (2019-02-27, branch: STABLE-2_1_x, SHA-1: 1bb6e7f52de1cd86be0eed31e3) to KSDK 2.0.0.
 - * Updated sys_thread_new implementation and comment.
 - * Kinetis ENET adaptation layer - reading frames into a pbuf chain is conditionally compiled only when a single pbuf from pool cannot hold maximum frame size (PBUF_POOL_BUFSIZE >= maximum frame size). Avoiding this code also reduces stack size requirements by about 1.5 kilobytes.
 - Bug fixes:
 - * Fixes in ethernetif_linkoutput() in enet_ethernetif_lpc.c:
 - Removed access to possibly freed pbuf.
 - Call pbuf_free() when transmit buffers not available.
 - When copying pbuf chain, updating the number of necessary transmit buffers to wait

- for, which can be often smaller in the copy.
- * When CGI script is reading POST data by chunks, the loop in `httpsrv_read()` may cause blocking in receive function waiting for more data at the end of the stream
 - `HTTPSRV_cgi_read()` - added limiting of the last chunk length according to content length to avoid undesired blocking
- * Applied AUTOIP patch <https://savannah.nongnu.org/patch/?9847> - with modification to support multiple network interfaces.
- * Fixed buffer overflow in `httpsrv` when application provided CGI script does not handle the whole content of POST request
- Removed `LwipMibCompiler` contrib application as it contained LGPL licensed files in `Sharp-SnmpLib`.
- 2.1.2_rev3
 - New features:
 - * `lwiperf` updated with UDP client/server support from the patch 9751 (<https://savannah.nongnu.org/patch/?9751>)
- 2.1.2_rev2
 - Bug fixes:
 - * Fixed `lwiperf_abort()` in `lwiperf.c` to correctly close connections and free resources
- 2.1.2_rev1
 - New features:
 - * Ported lwIP 2.1.2 (2018-11-22, SHA-1: 159e31b689577dbf69cf0683bbaffbd71fa5ee10) to KSDK 2.0.0.
 - * Ported lwIP-contrib 2.1.0 (2018-09-24, SHA-1: 35b011d4cf4c4b480f8859c456587a884ec9d287) to KSDK 2.0.0.
- 2.0.3_rev1
 - New features:
 - * Ported lwIP 2.0.3 (2017-09-15, SHA-1: 92f23d6ca0971a32f2085b9480e738d34174417b) to KSDK 2.0.0.
- 2.0.2_rev1
 - New features:
 - * Ported lwIP 2.0.2 (2017-03-13, SHA-1: c0862d60746e2d1ceae69af4c6f24e469570ecef) to KSDK 2.0.0.
- 2.0.0_rev3
 - New features:
 - * Ported lwIP 2.0.0 (2016-11-10, SHA-1: 216bf89491815029aa15463a18744afa04df58fe) to KSDK 2.0.0.
- 2.0.0_rev2
 - New features:
 - * Ported lwIP 2.0.0 RC2 (2016-08-08, SHA-1: b1dfd00f9233d124514a36a8c8606990016f2ad4) to KSDK 2.0.0.
- 2.0.0_rev1
 - New features:
 - * Ported lwIP 2.0.0 RC0 (2016-05-26) to KSDK 2.0.0.
 - * Changed lwIP bare-metal examples to use poll-driven approach instead of interrupt-driven one.

- 1.4.1_rev2
 - New features:
 - * Enabled critical sections in lwIP.
 - Bug fixes:
 - * Fixed default lwIP packet-buffer size to be able to accept a maximum size frame from the ENET driver.
 - * Fixed possible drop of multi-frame packets during transmission.
- 1.4.1_rev1
 - New features:
 - * Ported lwIP 1.4.1 to KSDK 2.0.0.

mbedTLS for MCUXpresso SDK

The current version of mbedTLS is based on mbed TLS 2.28.1 branch released 2021-7-11

- 2.28.1
 - New features:
 - * Ported mbedTLS 2.28.1 to SDK.
- 2.28.0
 - New features:
 - * Ported mbedTLS 2.28.0 to SDK.
- 2.27.0
 - New features:
 - * Ported mbedTLS 2.27.0 to SDK.
- 2.26.0
 - New features:
 - * Ported mbedTLS 2.26.0 to SDK.

2.16.6_rev7

- Bug fixes:
 - Corrected definition of global variable `g_isCryptoHWInitialized` to be only internal static variable in `sssapi_mbedtls.c` file.

2.16.6_rev6

- Bug fixes:
 - Adding `#ifdef` in `ecdsa.c` to remove warning: "function "derive_mpi" was declared but never referenced", when alternative implementation of ECDSA sign and verify is used and not used Deterministic ECDSA, then was `derive_mpi` function never used.

MOTOR_CONTROL for KSDK

Current version is 1.1.0

- 1.1.0

- Initial version.

Multicore SDK

The current version of Multicore SDK is 2.13.0

- 2.13.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.10.0
 - * eRPC generator (erpcgen) v.1.10.0
 - * Multicore Manager (MCMgr) v4.1.3
 - * RPMsg-Lite v5.0.0
 - New features:
 - * eRPC: MUPort adaptation to new supported SoCs.
 - * eRPC: Simplifying CI with installing dependencies using shell script, GitHub PR #267.
 - * eRPC: Using event for waiting for sock connection in TCP python server, formatting python code, C specific includes, GitHub PR #269.
 - * eRPC: Endianness agnostic update, GitHub PR #276.
 - * eRPC: Assertion added for functions which are returning status on freeing memory, GitHub PR #277.
 - * eRPC: Fixed closing arbitrator server in unit tests, GitHub PR #293.
 - * eRPC: Makefile updated to reflect the correct header names, GitHub PR #295.
 - * eRPC: Compare value length to used length() in reading data from message buffer, GitHub PR #297.
 - * eRPC: Add TCP_NODELAY option to python, GitHub PR #298.
 - * eRPC: Replace EXPECT_TRUE with EXPECT_EQ in unit tests, GitHub PR #318.
 - * eRPC: Adapt rpmsg_lite based transports to changed rpmsg_lite_wait_for_link_up() API parameters.
 - * eRPC, erpcgen: Better distinguish which file can and cannot be linked by C linker, GitHub PR #266.
 - * eRPC, erpcgen: Stop checking if pointer is NULL before sending it to the erpc_free function, GitHub PR #275.
 - * eRPC, erpcgen: Changed api to count with more interfaces, GitHub PR #304.
 - * erpcgen: Check before reading from heap the buffer boundaries, GitHub PR #287.
 - * erpcgen: Several fixes for tests and CI, GitHub PR #289.
 - * erpcgen: Refactoring erpcgen code, GitHub PR #302.
 - * erpcgen: Fixed assigning const value to enum, GitHub PR #309.
 - * erpcgen: Enable runTesttest_enumErrorCode_allDirection, serialize enums as int32 instead of uint32.
 - * MCMgr: mcmgr_mu_internal.c code adaptation to new supported SoCs.
 - * RPMsg-Lite: Improved debug check buffers implementation - instead of checking the pointer fits into shared memory check the presence in the VirtIO ring descriptors list.
 - * RPMsg-Lite: Timeout parameter added to rpmsg_lite_wait_for_link_up API function.
 - * RPMsg-Lite: VRING_SIZE is set based on number of used buffers now (as calculated

- in `vring_init`) - updated for all platforms that are not communicating to Linux `rpmsg` counterpart.
 - * `RPMsg-Lite`: Fixed wrong `RL_VRING_OVERHEAD` macro comment in `platform.h` files.
 - * `RPMsg-Lite`: Misra corrections.
- 2.12.0_imx93
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.9.1
 - * eRPC generator (erpcgen) v.1.9.1
 - * Multicore Manager (MCMgr) v4.1.2
 - * `RPMsg-Lite` v4.0.1
 - New features:
 - * `RPMsg-Lite`: Added porting layers for i.mx93 device.
- 2.12.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.9.1
 - * eRPC generator (erpcgen) v.1.9.1
 - * Multicore Manager (MCMgr) v4.1.2
 - * `RPMsg-Lite` v4.0.0
 - New features:
 - * eRPC: Construct the USB CDC transport, rather than a client, GitHub PR #220.
 - * eRPC: Fix premature import of package, causing failure when attempting installation of Python library in a clean environment, GitHub PR #38, #226.
 - * eRPC: Improve python detection in make, GitHub PR #225.
 - * eRPC: Fix several warnings with deprecated call in pytest, GitHub PR #227.
 - * eRPC: Fix freeing union members when only default need be freed, GitHub PR #228.
 - * eRPC: Fix making test under Linux, GitHub PR #229.
 - * eRPC: Assert costumizing, GitHub PR #148.
 - * eRPC: Fix corrupt clientList bug in TransportArbitrator, GitHub PR #199.
 - * eRPC: Fix build issue when invoking g++ with `-Wno-error=free-nonheap-object`, GitHub PR #233.
 - * eRPC: Fix inout cases, GitHub PR #237.
 - * eRPC: Remove `ERPC_PRE_POST_ACTION` dependency on return type, GitHub PR #238.
 - * eRPC: Adding NULL to ptr when codec function failed, fixing memcpy when fail is present during deserialization, GitHub PR #253.
 - * eRPC: MessageBuffer usage improvement, GitHub PR #258.
 - * eRPC: Get rid for serial and enum34 dependency (enum34 is in python3 since 3.4 (from 2014)), GitHub PR #247.
 - * eRPC: Several MISRA violations addressed.
 - * eRPC: Fix timeout for Freertos semaphore, GitHub PR #251.
 - * eRPC: Use of `rpmsg_lite_wait_for_link_up()` in `rpmsg_lite` based transports, GitHub PR #223.
 - * eRPC: Fix codec nullptr dereferencing, GitHub PR #264.
 - * erpcgen: Fix two syntax errors in erpcgen Python output related to non-encapsulated unions, improved test for union, GitHub PR #206, #224.

- * erpcgen: Fix serialization of list/binary types, GitHub PR #240.
- * erpcgen: Fix empty list parsing, GitHub PR #72.
- * erpcgen: Fix templates for malloc errors, GitHub PR #110.
- * erpcgen: Get rid of encapsulated union declarations in global scale, improve enum usage in unions, GitHub PR #249, #250.
- * erpcgen: Fix compile error:UniqueIdChecker.cpp:156:104:'sort' was not declared, GitHub PR #265.
- * MCMgr: Update mcmgr_stop_core_internal() implementations to set core state to kMCMGR_ResetCoreState.
- * RPMsg-Lite: Introduce new rpmsg_lite_wait_for_link_up() API function - this allows to avoid using busy loops in rtos environments, GitHub PR #21.
- * RPMsg-Lite: Adjust rpmsg_lite_is_link_up() to return RL_TRUE/RL_FALSE.
- 2.11.1
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.9.0
 - * eRPC generator (erpcgen) v.1.9.0
 - * Multicore Manager (MCMgr) v4.1.1
 - * RPMsg-Lite v3.2.1
 - New features:
 - * RPMsg-Lite: Add support for custom shared memory arrangement per the RPMsg_Lite instance.
- 2.11.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.9.0
 - * eRPC generator (erpcgen) v.1.9.0
 - * Multicore Manager (MCMgr) v4.1.1
 - * RPMsg-Lite v3.2.0
 - New features:
 - * eRPC: Improving template usage, GitHub PR #153.
 - * eRPC: run_clang_format.py cleanup, GitHub PR #177.
 - * eRPC: Build TCP transport setup code into liberpc, GitHub PR #179.
 - * eRPC: Fix multiple definitions of g_client error, GitHub PR #180.
 - * eRPC: Fix memset past end of buffer in erpc_setup_mbf_static.cpp, GitHub PR #184.
 - * eRPC: Fix deprecated error with newer pytest version, GitHub PR #203.
 - * eRPC: Allow used LIBUSB_SIO device index being specified from the Python command line argument.
 - * eRPC, erpcgen: Static allocation support and usage of rpmsg static FreeRTOSs related API, GitHub PR #168, #169.
 - * erpcgen: Remove redundant module imports in erpcgen, GitHub PR #196.
 - * RPMsg-Lite: Improve static allocations - allow OS-specific objects being allocated statically, GitHub PR #14.
 - * RPMsg-Lite: Minor Misra and typo corrections, GitHub PR #19, #20.
- 2.10.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.8.1

- * eRPC generator (erpcgen) v.1.8.1
- * Multicore Manager (MCMgr) v4.1.1
- * RPMsg-Lite v3.1.2
- New features:
 - * eRPC: Fix misra erpc c, GitHub PR #158.
 - * eRPC: Allow conditional compilation of message_loggers and pre_post_action.
 - * eRPC: New i2c_slave_transport transport introduced.
 - * eRPC: (D)SPI slave transports updated to avoid busy loops in rtos environments.
 - * erpcgen: Re-implement EnumMember::hasValue(), GitHub PR #159.
 - * erpcgen: Fixing several misra issues in shim code, erpcgen and unit tests updated, GitHub PR #156.
 - * erpcgen: Fix bison file, GitHub PR #156.
 - * RPMsg-Lite: Fixed incorrect description of the rpmsg_lite_get_endpoint_from_addr function.
 - * RPMsg-Lite: Updated RL_BUFFER_COUNT documentation.
 - * RPMsg-Lite: env_print macro adjusted to address MISRA 21.6 rule in MCUXpressoSDK projects.
- 2.9.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.8.0
 - * eRPC generator (erpcgen) v.1.8.0
 - * Multicore Manager (MCMgr) v4.1.1
 - * RPMsg-Lite v3.1.1
 - New features:
 - * eRPC: Support win32 thread, GitHub PR #108.
 - * eRPC: Add mbed support for malloc() and free(), GitHub PR #92.
 - * eRPC: Update makefile.
 - * eRPC: Fixed warnings and error with using MessageLoggers, GitHub PR #127.
 - * eRPC: Extend error msg for python server service handle function, GitHub PR #132.
 - * eRPC: Update CMSIS UART transport layer to avoid busy loops in rtos environments, introduce semaphores.
 - * eRPC: Introduced pre and post callbacks for eRPC call, GitHub PR #131.
 - * eRPC: Introduced new USB CDC transport.
 - * eRPC: Introduced new Linux spidev-based transport.
 - * eRPC: SPI transport update to allow usage without handshaking GPIO.
 - * eRPC: Native WIN32 *erpc serial transport and threading*.
 - * *eRPC: Arbitrator deadlock fix, TCP transport updated, TCP setup functions introduced, GitHub PR #121.*
 - * *eRPC: Update of matrix_multiply.py example: Add –serial and –baud argument, GitHub PR #137.*
 - * *eRPC: Added formatting extension for VSC, GitHub PR #134.*
 - * *eRPC: Update of .clang-format, GitHub PR #140.*
 - * *eRPC: Update of erpc_framed_transport.cpp: return error if received message has zero length, GitHub PR #141.*
 - * *eRPC, erpcgen: Fixed error messages produced by -Wall -Wextra -Wshadow -pedantic-*

errors compiler flags, GitHub PR #136, #139.

- * *eRPC, erpcgen: Core re-formatted using Clang version 10.*
 - * *erpcgen: Enable deallocation in server shim code when callback/function pointer used as out parameter in IDL.*
 - * *erpcgen: Removed '\$' character from generated symbol name in '\$union' suffix, GitHub PR #103.*
 - * *erpcgen: Resolved mismatch between C++ and Python for callback index type, GitHub PR #111.*
 - * *erpcgen: Python generator improvements, GitHub PR #100, #118.*
 - * *erpcgen: Fixed error messages produced by -Wall -Wextra -Wshadow -pedantic-errors compiler flags, GitHub PR #136.*
 - * *erpcgen: Introduce ustring type for unsigned char and force cast to char*, GitHub PR #125.*
 - * *RPMMsg-Lite: Introduced RL_ALLOW_CONSUMED_BUFFERS_NOTIFICATION config option to allow opposite side notification sending each time received buffers are consumed and put into the queue of available buffers.*
 - * *RPMMsg-Lite: Added environment layers for Threadx.*
- 2.8.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.7.4
 - * eRPC generator (erpcgen) v.1.7.4
 - * Multicore Manager (MCMgr) v4.1.0
 - * RPMMsg-Lite v3.1.0
 - New features:
 - * eRPC: Unit test code updated to handle service add and remove operations.
 - * eRPC: Several MISRA issues in rpmsg-based transports addressed.
 - * eRPC: Support MU transport unit testing.
 - * eRPC: Adding mbed os support.
 - * eRPC: Fixed Linux/TCP acceptance tests in release target.
 - * eRPC: Minor documentation updates, code formatting.
 - * erpcgen: Whitespace removed from C common header template.
 - * RPMMsg-Lite: MISRA C-2012 violations fixed (7.4).
 - * RPMMsg-Lite: Fix missing lock in rpmsg_lite_rx_callback() for QNX env.
 - * RPMMsg-Lite: Correction of rpmsg_lite_instance structure members description.
 - * RPMMsg-Lite: Address -Waddress-of-packed-member warnings in GCC9.
 - * RPMMsg-Lite: Clang update to v10.0.0, code re-formatted.
 - 2.7.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.7.3
 - * eRPC generator (erpcgen) v.1.7.3
 - * Multicore Manager (MCMgr) v4.1.0
 - * RPMMsg-Lite v3.0.0
 - New features:
 - * eRPC: Improved the test_callbacks logic to be more understandable and to allow requested callback execution on the server side.

- * eRPC: TransportArbitrator::prepareClientReceive modified to avoid incorrect return value type.
- * eRPC: The ClientManager and the ArbitratedClientManager updated to avoid performing client requests when the previous serialization phase fails.
- * erpcgen: Generate the shim code for destroy of statically allocated services.
- * MCMgr: Code adjustments to address MISRA C-2012 Rules
- * RPMsg-Lite: MISRA C-2012 violations fixed, incl. data types consolidation.
- * RPMsg-Lite: Code formatted
- 2.6.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.7.2
 - * eRPC generator (erpcgen) v1.7.2
 - * Multicore Manager (MCMgr) v4.0.3
 - * RPMsg-Lite v2.2.0
 - New features:
 - * eRPC: Improved support of const types.
 - * eRPC: Fixed Mac build.
 - * eRPC: Fixed serializing python list.
 - * eRPC: Documentation update.
 - * eRPC: Add missing doxygen comments for transports.
 - * RPMsg-Lite: Added configuration macro RL_DEBUG_CHECK_BUFFERS.
 - * RPMsg-Lite: Several MISRA violations fixed.
 - * RPMsg-Lite: Added environment layers for QNX and Zephyr.
 - * RPMsg-Lite: Allow environment context required for some environments (controlled by the RL_USE_ENVIRONMENT_CONTEXT configuration macro).
 - * RPMsg-Lite: Data types consolidation.
 - * MCMgr: Documentation updated to describe handshaking in a graphic form.
 - * MCMgr: Minor code adjustments based on static analysis tool findings
- 2.5.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.7.1
 - * eRPC generator (erpcgen) v1.7.1
 - * Multicore Manager (MCMgr) v4.0.2
 - * RPMsg-Lite v2.0.2
 - New features:
 - * RPMsg-Lite, MCMgr: Align porting layers to the updated MCUXpressoSDK feature files.
 - * eRPC: Fixed semaphore in static message buffer factory.
 - * erpcgen: Fixed MU received error flag.
 - * erpcgen: Fixed tcp transport.
- 2.4.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.7.0
 - * eRPC generator (erpcgen) v1.7.0
 - * Multicore Manager (MCMgr) v4.0.1
 - * RPMsg-Lite v2.0.1

- New features:
 - * eRPC: Improved code size of generated code.
 - * eRPC: Generating crc value is optional.
 - * eRPC: Fixed CMSIS Uart driver. Removed dependency on KSDK.
 - * eRPC: List names are based on their types. Names are more deterministic.
 - * eRPC: Service objects are as a default created as global static objects.
 - * eRPC: Added missing doxygen comments.
 - * eRPC: Forbid users use reserved words.
 - * eRPC: Removed outByref for function parameters.
 - * eRPC: Added support for 64bit numbers.
 - * eRPC: Added support of program language specific annotations.
 - * eRPC: Optimized code style of callback functions.
 - * RMPMsg-Lite: New API rmpmsg_queue_get_current_size()
 - * RMPMsg-Lite: Fixed bug in interrupt handling for lpc5411x, lpc5410x
 - * RMPMsg-Lite: Code adjustments based on static analysis tool findings
- 2.3.1
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.6.0
 - * eRPC generator (erpcgen) v.1.6.0
 - * Multicore Manager (MCMgr) v4.0.0
 - * RMPMsg-Lite v1.2.0
 - New features:
 - * eRPC: Improved code size of generated code.
 - * eRPC: Improved eRPC nested calls.
 - * eRPC: Improved eRPC list length variable serialization.
 - * eRPC: Added @nullable support for scalar types.
 - * MCMgr: Added new MCMGR_TriggerEventForce() API.
- 2.3.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.5.0
 - * eRPC generator (erpcgen) v.1.5.0
 - * Multicore Manager (MCMgr) v3.0.0
 - * RMPMsg-Lite v1.2.0
 - New features:
 - * eRPC: Added support for unions type non-wrapped by structure.
 - * eRPC: Added callbacks support.
 - * eRPC: Added support @external annotation for functions.
 - * eRPC: Added support @name annotation.
 - * eRPC: Added Messaging Unit transport layer.
 - * eRPC: Added RMPMSG Lite RTOS TTY transport layer.
 - * eRPC: Added version verification and IDL version verification between eRPC code and eRPC generated shim code.
 - * eRPC: Added support of shared memory pointer.
 - * eRPC: Added annotation to forbid generating const keyword for function parameters.
 - * eRPC: Added python matrix multiply example.

- * eRPC: Added nested call support.
- * eRPC: Added struct member "byref" option support.
- * eRPC: Added support of forward declarations of structures
- * eRPC: Added Python RPMMsg Multiendpoint kernel module support
- * eRPC: Added eRPC sniffer tool
- * MCMgr: Unused API removed
- * MCMgr: Added the ability for remote core monitoring and event handling
- * RPMMsg-Lite: Several source files renamed to avoid conflicts with other middleware sw components
- * RPMMsg-Lite: Added the ability to use Multicore Manager (MCMGR) as the IPC interrupts router
- 2.2.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.4.0
 - * eRPC generator (erpcgen) v.1.4.0
 - * Multicore Manager (MCMgr) v2.0.1
 - * RPMMsg-Lite v1.1.0
 - New features:
 - * eRPC: win_flex_bison.zip for windows updated.
 - * eRPC: Use one codec (instead of inCodec outCodec).
 - * eRPC: New RPMMsg-Lite Zero Copy (RPMMsgZC) transport layer.
 - * MCMgr: code updated to be Misra compliant.
 - * RPMMsg-Lite: Added macros for packed structures (compiler.h).
 - * RPMMsg-Lite: Improved interrupt handling in platform layer.
 - * RPMMsg-Lite: Changed RL_BUFFER_SIZE definition.
 - * RPMMsg-Lite: Fix of double initialization of vring shared data structure.
 - * RPMMsg-Lite: Support for the multi-instance.
- 2.1.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.3.0
 - * eRPC generator (erpcgen) v.1.3.0
 - New features:
 - * eRPC: New annotation types introduced (@length, @max_length, ...).
 - * eRPC: Support for running both erpc client and erpc server on one side.
 - * eRPC: New transport layers for (LP)UART, (D)SPI.
 - * eRPC: Error handling support.
- 2.0.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.2.0
 - * eRPC generator (erpcgen) v.1.2.0
 - * Multicore Manager (MCMgr) v2.0.0
 - * RPMMsg-Lite v1.0.0
 - New features:
 - * Multicore SDK support for lpcxpresso54114 board added.
 - * RPMMsg component of the Open-AMP framework re-implemented and the RPMMsg-Lite

- version introduced.
- * eRPC source directory organization changed.
- * Many eRPC improvements.
- 1.1.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.1.0
 - * Multicore Manager (MCMgr) v1.1.0
 - * Open-AMP / RPMsg based on SHA1 ID 44b5f3c0a6458f3cf80 rev01
 - New features:
 - * Multicore SDK 1.1.0 ported to KSDK 2.0.0.
 - * Python support added into eRPC.
- 1.0.0
 - Multicore SDK component versions:
 - * embedded Remote Procedure Call (eRPC) v1.0.0
 - * Multicore Manager (MCMgr) v1.0.0
 - * Open-AMP / RPMsg based on SHA1 ID 44b5f3c0a6458f3cf80 rev00

NTAG I2C plus library

The current version is 1.0.0.

- 1.0.0
 - initial release.

RTCESL for KSDK

Current version is 4.3

- 4.3
 - Initial version.

SAFETY_IEC60730B for KSDK

Current version is 1.1.0

- 1.1.0
 - Initial version.

Host SDIF driver for MCUXpresso SDK

The current driver version is 2.4.1.

- 2.4.1

- Improvements
 - * Added macro SDMMCHOST_SUPPORT_VOLTAGE_CONTROL.
- 2.4.0
 - Improvements
 - * Removed deprecated api in SDIF host driver.
 - * Added SDMMCHOST_ConvertDataToLittleEndian api.
 - * Added capability/maxBlockCount/maxBlockSize in host decription.
 - * Added mutual exclusive access for function init/deinit/reset/transfer function.
 - * Fixed violations of MISRA C-2012 rule 10.1.
- 2.3.1
 - Improvements
 - * Added host instance capability macro.
 - * Added clear card inserted/removed event when card removed/inserted interrupt generated.
 - * Increased the reset timeout value to fix the data machine still busy after sdif reset issue.
 - * Enabled the error recovery function by adding host reset operations.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.3.0
 - Improvements
 - * Merged the host controller driver from polling/freertos/interrupt to non_blocking/blocking.
 - * Added SDMMC OSA layer to support muxtex access/event/delay.
- 2.2.14
 - Bug Fixes
 - * Fixed uninitialized value Coverity issue.
- 2.2.13
 - Improvements:
 - * Added host reset after the card being powered on for host controller SDIF to fix the DATA_BUSY issue.
 - * Removed the SDIF_Reset from SDMMCHOST_Reset.
- 2.0.0
 - Initial version

MMC Card driver for MCUXpresso SDK

The current driver version is 2.5.0.

- 2.5.0
 - Improvements
 - * Added api MMC_SetSleepAwake to support enter/exit sleep state.
 - * Added new api MMC_PollingCardStatusBusy for application polling card status.
 - * Removed deprecated api in mmc driver and mark MMC_HostReset as deprecated.
 - * Improved the read/write/erase function flow.
 - * Added mutual exclusive access for init/deinit/read/write/erase function.

- * Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.7, 10.4, 13.5, 14.4, 10.6.
- 2.4.1
 - Improvements
 - * Improved the voltage window argument of CMD1 according to host capability instead of use card ocr directly.
 - * Added host HS200/HS400/8bit bus width capability validation during card initialization.
 - * Used cache line size align buffer for MMC relate api.
 - * Increased the CMD13 timeout count to avoid polling CMD13 time out issue.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.4.0
 - Improvements
 - * Added new apis MMC_EnableCacheControl/MMC_FlushCache to support cache feature.
- 2.3.1
 - Improvements
 - * Removed the dead loop while polling DAT0 and CMD13 instead of using timeout mechanism.
 - * Added card state check before switching to HS400 to improve the emmc initialization stability.
 - * Removed the redundant operation of memset internal buffer in MMC_WrtiteBlocks function.
 - Bug Fixes
 - * Fixed the sandisk emmc always busy while sending CMD1 without supported voltage provide in argument.
- 2.3.0
 - Improvements
 - * Deprecated api MMC_PowerOnCard/MMC_PowerOffCard by api MMC_SetCardPower.
 - * Added internalBuffer in mmc_card_t and removed rawCid/rawCsd/rawExtendedCsd.
 - * Added retuning support during data transfer under HS200 mode.
 - * Increased the read/write blocks failed retry times for stability.
 - * Added delay while retry the CMD1 for stability.
 - * Added legacy card support, the card not support CMD6, CMD8.
- 2.2.13
 - Improvements
 - * Used the boot mode value instead of boot mode mask value as the parameter of MMC_SetBootConfig to improve user experience.
 - * Removed dynamic voltage switch feature for mmc, according to JEDEC standard, the voltage should be fixed after power up.
- 2.2.12
 - Improvement
 - * Increased the CMD1 retry times in the MMC card driver to improve driver compatibility.
 - Bug Fixes
 - * Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).

- * Fixed the fall through build warning by adding SUPPRESS_FALL_THROUGH_WARNING() in mmc driver.
- 2.2.7
 - Bug Fixes
 - * Fixed MDK 66-D warning.
- 2.2.6
 - Improvements
 - * Saved MMC OCR registers while sending CMD1 with argument 0.
 - Bug Fixes
 - * Added MMC_PowerOn function in which there is delay function after powerup sdcard. Otherwise, the card initialization by fail.
- 2.2.5
 - Improvements
 - * Added SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning and it is disabled by default.
- 2.2.4
 - Bug Fixes
 - * Fixed DDR mode data sequence miss issue, which is caused by NIBBLE_POS.
 - Improvements
 - * Increased g_sdmmc 512byte to improve the performance when application use a non-word align data buffer address.
 - * Used OCR access mode bits to determine the mmccard high capacity flag.
- 2.2.3
 - Bug Fixes
 - * Added response check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.1
 - Improvements
 - * Improved MMC Boot feature.
- 2.2.0
 - Improvements
 - * Optimized tuning/mmc switch voltage/mmc select power class/mmc select timing function.
 - * Added strobe dll for mmc HS400 mode.
 - * Added write complete wait operation for MMC_Write to fix command timeout issue.
- 2.1.2
 - Improvements
 - * Improved SDMMC to support eMMC v5.0.
 - Bug Fixes
 - * Fixed incorrect comparison between count and length in MMC_ReadBlocks/MMC_WriteBlocks.
- 2.1.1
 - Bug Fixes
 - * Fixed the block range boundary error when transferring data to MMC card.
- 2.1.0

- Improvements
 - * Optimized the function of setting maximum data bus width for MMC card.
- 2.0.0
 - Initial version

SD Card driver for MCUXpresso SDK

The current driver version is 2.4.1.

- 2.4.1
 - Improvements
 - * Added macro `SDMMCHOST_SUPPORT_VOLTAGE_CONTROL` for the host which not support voltage control.
- 2.4.0
 - Improvements
 - * Removed deprecated api in sd driver.
 - * Added new api `SD_PollingCardStatusBusy` for application polling card status.
 - * Improved the read/write/erase function flow.
 - * Improved the signal line voltage switch flow.
 - * Added `powerOnDelayMS/powerOffDelayMS` in `sd_usr_param_t` to allow redefine the default power on/off delay.
 - * Added mutual exclusive access for init/deinit/read/write/erase function.
 - * Fixed the driver strength configurations missed when timing mode switch to non SDR50/-SDR104 mode.
 - * Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.7, 10.4, 13.5, 14.4.
- 2.3.3
 - Improvements
 - * Added host SDR timing mode capability validation during card initialization.
 - * Added polling card ready for data status when transfer data failed.
 - * Used cache line size align buffer for SD initialization api.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.3.2
 - Improvements
 - * Moved power off function after card detect in `SD_Init` for DAT3 detect card feature.
- 2.3.1
 - Improvements
 - * Removed the dead loop while polling DAT0 and CMD13 instead of using timeout mechanism.
- 2.3.0
 - Improvements
 - * Marked api `SD_HostReset/SD_PowerOnCard/SD_PowerOffCard/SD_WaitCardDetect-Status` as deprecated.

- * Added new api SD_SetCardPower/SD_PollingCardDetectStatus/SD_HostDoReset.
- * Added internalBuffer in sd_card_t and removed rawCid/rawCsd/rawScr.
- * Added retuning support during data transfer under SDR50/SDR104 mode.
- * Increased the read/write blocks failed retry times for stability.
- * Added delay while retry the ACMD41 for stability.
- 2.2.12
 - Improvements
 - * Increased the sd io driver strength for SD2.0 card.
 - Bug Fixes
 - * Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).
- 2.2.10
 - Bug Fixes
 - * Added event value check for all the FreeRTOS events to fix program hangs when a card event occurs before create.
- 2.2.7
 - Bug Fixes
 - * Fixed MDK 66-D warning.
- 2.2.5
 - Improvements
 - * Added SD_ReadStatus api to get 512bit SD status.
 - * Added error log support in sdcard functions.
 - * Added SDMMC_ENABLE_SOFTWARE_TUNING to enable/disable software tuning and it is disabled by default.
- 2.2.4
 - Bug Fixes
 - * Fixed DDR mode data sequence miss issue, which is caused by NIBBLE_POS.
 - Improvements
 - * Increased g_sdmmc 512byte to improve the performance when application use a non-word align data buffer address.
 - * Enabled auto cmd12 for SD read/write.
- 2.2.3
 - Bug Fixes
 - * Added response check for send operation condition command. If not checked, the card may occasionally init fail.
- 2.2.1
 - Improvements
 - * Kept SD_Init function for forward compatibility.
- 2.2.0
 - Improvements
 - * Separated the SD/MMC/SDIO init API to xxx_CardInit/xxx_HostInit.
 - * SD_Init/SDIO_Init will be deprecated in the next version.
- 2.1.6
 - Improvements
 - * Enhanced SD IO default driver strength.

- 2.1.5
 - Bug Fixes
 - * Fixed Coverity issue.
 - * Fixed SD v1.x card write fail issue. It was caused by the block length set error.
 - * Fixed card cannot detect dynamically.
- 2.1.3
 - Bug Fixes
 - * Fixed Non high-speed sdcard init fail at switch to high speed.
 - Improvements
 - * Added Delay for SDCard power up.
- 2.1.2
 - Improvements
 - * Improved SDMMC to support SD v3.0.
- 2.1.1
 - Bug Fixes
 - * Fixed the bit mask error in the SD card switch to high speed function.
 - Improvements
 - * Optimized the SD card initialization function.
- 2.1.0
 - Bug Fixes
 - * Changed the callback mechanism when sending a command.
 - * Fixed the performance low issue when transferring data.
 - Improvements
 - * Changed the name of some error codes returned by internal function.
 - * Merged all host related attributes to one structure.
- 2.0.0
 - Initial version.

SDIO Card driver for MCUXpresso SDK

The current driver version is 2.4.1.

- 2.4.1
 - Improvements
 - * Added macro SDMMCHOST_SUPPORT_VOLTAGE_CONTROL for the host which not support voltage control.
- 2.4.0
 - Improvements
 - * Removed deprecated api in sdio driver.
 - * Improved the signal line voltage switch flow.
 - * Added powerOnDelayMS/powerOffDelayMS in sdio_usr_param_t to allow redefine the default power on/off delay.
 - * Added mutual exclusive access for init/deinit/direct/extend function.
 - * Fixed violations of MISRA C-2012 rule 4.7, 17.7, 10.1, 12.2.

- 2.3.3
 - Bug Fixes
 - * Fixed logical dead code coverity issue.
 - Improvements
 - * Removed deprecated api in sdio driver.
- 2.3.2
 - Improvements
 - * Added host SDR timing mode capability validation during card initialization.
 - * Used cache line size align buffer for SDIO initialization api.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 11.9, 15.7, 4.7, 16.4, 10.1, 10.3, 10.4, 11.3, 14.4, 10.6, 17.7, 16.1, 16.3.
- 2.3.1
 - Improvements
 - * Moved power off function after card detect in SD_Init for DAT3 detect card feature.
- 2.3.0
 - Improvements
 - * Marked api SDIO_HostReset/SDIO_PowerOnCard/SDIO_PowerOffCard/SDIO_Wait-CardDetectStatus as deprecated.
 - * Added new api SDIO_SetCardPower/SDIO_PollingCardDetectStatus/SDIO_HostDo-Reset.
 - * Added internalBuffer in sdio_card_t for card register content extract and improve the data access efficiency.
 - * Added retry function after switch to target timing failed in SDIO_SelectBusTiming.
 - * Changed default bus clock from 400KHZ to 25MHZ.
- 2.2.13
 - Improvements
 - * Removed the sdio card interrupt from sdio host initialization, since the card interrupt enablement should be determined by application.
 - Bug Fixes
 - * Fixed Out-of-bounds write Coverity issue.
- 2.2.12
 - Improvements
 - * Added manual tuning function for looking for the tuning window automatically.
 - * Fixed the build warning by changing the old style function declaration static status_t inline to static inline status_t(found by adding -Wold-style-declaration in armgcc build flag).
 - * Fixed the fall through build warning by adding SUPPRESS_FALL_THROUGH_WARNING() in sdio driver.
- 2.2.11
 - Bug Fixes
 - * Added check card async interrupt capability in function SDIO_GetCardCapability.
 - * Fixed OUT OF BOUNDS access in function SDIO_IO_Transfer.
- 2.2.10
 - Bug Fixes
 - * Fixed SDIO card driver get an incorrect io number when the card io number is bigger than

- 2.
 - Improvements
 - * Added SDIO 3.0 support.
 - * Added API SDIO_IO_RW_Direct for direct read/write card register access.
- 2.2.9
 - Improvements
 - * Added API SDIO_SetIOIRQHandler/SDIO_HandlePendingIOInterrupt to handle multi io pending IRQ.
- 2.2.8
 - Improvements
 - * Updated sdmmc to support SDIO interrupt.
 - * Added API SDIO_GetPendingInterrupt to get the pending io interrupt.
- 2.2.7
 - Bug Fixes
 - * Fixed MDK 66-D warning.
- 2.2.6
 - Improvements
 - * Added an unify transfer interface for SDIO.
 - Bug Fixes
 - * Fixed Wrong pointer address used by SDMMCHOST_Init.
- 2.1.5
 - Improvements
 - * Improved SDIO card init sequence and add retry option for SDIO_SwitchToHighSpeed function.
- 2.1.4
 - Improvements
 - * Added Go_Idle function for SDIO card.
- 2.0.0
 - Initial version.

Secure Element hostlib for KSDK

- For Detailed change log please download the Plug & Trust MW package from https://www.nxp.com/products/:SE050?tab=Design_Tools_Tab and refer to the Changes Present in the User guide.
- 3.01.00
 - Cleanup for heap management macros. Added support to redirect macros to FreeRTOS APIs.
- 3.00.03
 - smCom_Init: return type is now *UI6* instead of *void*. Return value indicates success/failure to create mutex/semaphore.
- 3.00.02
 - Fixed: potential null pointer dereference
- 2.16.00
 - MIMXRT1050 support replaced with MIMXRT1060

- Examples renamed
- 2.14.00
 - Extensively revamped `fsl_sss_ftr.h` file for finer control of build configuration selection.
 - smCom Layer is refactored so that Application send down the connection handles/parameters to lower layer.
 - Define `T10I2C_UM1225_SE050` is no longer applicable, use `T10I2C_UM11225_SE05X` instead.
- 2.11.03
 - hostlib file/folder moved
- 2.11.0
 - Added Support for SE050 and SSS API's
- 1.6.0
 - Host Library version goes from 01.40 to 01.41 (A71CH Host API has not been extended)
 - Conditional translation of `i2c_Failed` into `i2c_NoAddrAck` removed (this translation is no longer required as a NACK is no longer lumped into `i2c_Failed` by the I2C driver wrapper)
 - Additional update of SCI2C implementation: Ensure multiple consecutive NACK's on address will trigger a return from function '`sci2c_SlaveToMasterDataTx`'.
 - `sci2c_Init` update: initial loop fetching `sci2c` status modified so a SoftReset is issued for all exception types.
 - Solved potential buffer overflow in the implementation of `smAduAppendCmdData`
 - Added additional explicit checks that pointer arguments passed at the `AX_API/A71CH_API` level are not NULL pointers. In case this check was done through an '`assert`' statement, the '`assert`' statement has been removed.
- 1.5.0
 - Added i.MX RT1050 EVKB and LPC54018 IoT module projects
 - Added Watson IoT demo
 - GP Storage Lookup table is no longer restricted to 5 objects, up to 254 objects can be stored (use `HLSE_MAX_OBJECTS_IN_TABLE (>=8)` to limit support and save memory)
 - GP Storage Update Counter can be disabled by defining `HLSE_DISABLE_UPDATE_COUNTER` (not defined by default).
 - It's possible to fetch the absolute storage offset of an object
 - Specific error code when attempting to update a locked Lookup table
 - Extended doxygen documentation
- 1.4.2
 - Upgraded `rtos\amazon-freertos\lib\pkcs11\portable\nxp\se_hostlib\pkcs11_se.c`
 - Added GCP (Google Cloud Platform Demo)
 - Readme updates for AWS JITR Demo
- 1.4.0.1
 - Examples: Fix time stamp printing on embedded platform. (Time stamps are supported only for Linux/Windows)
- 1.0.0
 - New integration

Trusted Firmware M (TF-M) for MCUXpresso SDK

The current version is based on TF-M v1.6.0, released 2022-11-07 (50c3bb5218a6ba37316f1507e4220129b2d0edf6) on <https://git.trustedfirmware.org/TF-M/trusted-firmware-m.git>

- 1.6.0 Rev1
 - Ported TF-M v1.6.0 to MCUXpresso SDK. Based on the 2022-11-07 snapshot(50c3bb5218a6ba37316f1507e4220129b2d0edf6)
 - Added lpcxpresso55s36 platform.
- 1.6.0
 - Ported TF-M v1.6.0 to MCUXpresso SDK. Based on the 2022-08-29 snapshot(a5048fb687e4fbf28f9aac69)
 - Added lpcxpresso55s36 platform.
- 1.5.0
 - Ported TF-M v1.5.0 to MCUXpresso SDK. Based on the 2022-03-11 snapshot(47c26ecd0dec177fe2ddf82)
- 1.4.0
 - Ported TF-M v1.4.0 to MCUXpresso SDK. Based on the 2021-08-30 snapshot(00c1106624d733aade4486)
- 1.3.0
 - Ported TF-M v1.3.0 to MCUXpresso SDK. Based on the 2021-04-13 snapshot(cad01ab98c34bb3a13ab49c)
 - Added evkmimxrt685 platform.
 - Added evkmimxrt595 platform.
- 1.1
 - Ported TF-M v1.1 to MCUXpresso SDK. Based on the 2020-08-19 snapshot(54507b1645087e92b5a11591)
- 1.0 Rev3
 - Ported TF-M v1.0 to MCUXpresso SDK. Based on the 2020-04-20 snapshot(1e089705899ff68c617cc0c01)
 - Added HUK derivation function.
- 1.0 Rev2
 - Ported TF-M v1.0 to MCUXpresso SDK. Based on the 2020-04-06 snapshot(d4ac5d14e74d1aa2a78526aa)
 - Added the LPC55S Flash module support for ITS and PS.
 - Added lpcxpresso55s16 platform.
- 1.0 Rev1
 - Ported TF-M v1.0-RC1 to MCUXpresso SDK. Based on the 2019-09-23 snapshot(011c0ad0e76d62bd6f2d)
 - Added MCUX and GCC demo applications.
 - Added PSA test suite application.
 - Used mbedCrypto instead of mbedTLS
- 1.0
 - Ported TF-M v1.0-beta to MCUXpresso SDK. Based on the 2019-03-13 snapshot(a5a2a5bc32bc50f4def8d)
 - Added lpcxpresso55s69 platform.
 - Added MDK demo applications.
 - Bug Fixes:
 - * Fixed compilation warnings and errors.
 - * Changes in the TF-M original source code are marked by the "NXP" comment.

USB stack for MCUXpresso SDK

The current version of USB stack is 2.8.4.

- 2.8.4
 - Improvement:
 - * Add the new netc adapter for the new netc driver.
 - * Fix issues for USB device dfu and usb device msc when enable the macro USB_DEVICE_CONFIG_RETURN_VALUE_CHECK.
 - * Change the header file including order for usb.h header.
 - * Update the USB host audio class driver to fix the wrong output log.
 - * Add the workaround on dev_hid_mouse_bm case for the errata TN00071.
 - * Enable ROOT2 macro in USB device stack.
 - * Use an unified definition for the base address of RTxxxx platforms.
- 2.8.3
 - Improvement:
 - * Update the EHCI controller driver to support the address convert for TCM.
 - * Update the USB host EHCI controller driver to make sure the mutual exclusion access under multiple tasks' environment.
- 2.8.2
 - Improvement:
 - * Fix noise issue of UAC 3.1, UAC 5.1, UAC 7.1 on usb audio speaker demo.
 - * Fix the issue that incorrect PC behavior when ejecting USB MSC devices.
 - * Update the EHCI controller driver to support RW610 that does not reply on PHY driver, especially for low power feature.
 - * Update the USB_HostHelperParseAlternateSetting to fix the wrong interface parse.
 - * Update dev_composite_hid_audio_unified_bm demo to support independent mute/unmute and volume control.
- 2.8.1
 - Improvement:
 - * update USB audio demos to use audio component (components).
 - * Add the checking of function call return value.
 - * Add audio multiple channels demo (usb_device_composite_audio_multi_ch_unified) on RT600 audio board.
 - * Fix audio noise on sync mode and improve overflow/underflow checking method.
 - * Support UAC 3.1, 5.1 and 7.1 on audio speaker demo.
 - * Set USB device CDC demo not to depend on DTR setting from host.
 - * Support MCUX toolchain on some RTxxxx platforms.
- 2.8.0
 - Improvement:
 - * Fix the USB device stack vulnerability issues.
 - * Update the audio PLL and FRO adjustment codes for audio examples in RTxxx, LPC54xxx and LPC55xxx.
 - * Improve the USB PD AMS collision avoidance.
 - * Improve IP3511 controller driver's dedicated ram allocation.
 - * Change the USB_DATA_ALIGN_SIZE to 4 because the controller driver uses the dedicated RAM to do memcpy.
 - New features:
 - * Enable USB host audio recorder demo for multiple boards.

- 2.7.0
 - Improvement:
 - * Use new feedback solution and low latency playback for usb device speaker demo and unified demos. Add underflow and overflow protection.
 - * Optimize hard code for usb audio demos.
 - * Update Unconstrained Power field in the Sink Capabilities Message according to the external power state.
 - * Fix CVE-2021-38258 and CVE-2021-38260
 - New features:
 - * Enable USB host video demo for mutilple boards.
 - * Enable USB device MTP demo for mutilple boards.
 - * Add PPS message to usb pd stack.
- 2.6.1
 - Improvement:
 - * rename sdcard as disk for all of sdcard demos. For ramdisk demos, they are not changed.
 - * add wrapper for all of disk demos to support emmc.
- 2.6.0
 - Improvement:
 - * Added more ufi event to support dynamic sdcard capacity.
 - * Passed MISRA-2012 mandatory and required rules.
 - Except rule 17.2 in host hub and otg stack.
 - Except rule 5.1, rule 5.4, rule 21.1 and rule 21.2.
 - * Re-implemented USB components and supported NPW.
 - * Improved IP3511 controller driver's cancelling transfer function.
 - * Enabled the audio2.0 defaultly for device audio demos.
 - * Enabled the host audio2.0 function in host audio class driver and host audio speaker demo.
 - New features:
 - * enable two USB controllers in one USB host mouse demo which named as host_hid_-mouse_dual.
 - * enable UAC 5.1 for usb device audio speaker demo.
- 2.5.0
 - Improvement:
 - * Integrated sdk components (OSA, Timer, GPIO and serial_manager) to USB stack and demos.
 - * Improved the ip3511 driver throughput.
 - * Improved audio initialization codes after SDK audio drivers update.
 - * Improved audio to support the audio2.0 in win10.
 - * Add one "enumeration fail" callback event to host stack.
- 2.4.2
 - Improvement:
 - * Put the USB controller data and transfer buffer to noncache section, removed the setting that sets the whole ocram and sdram as noncached.
 - * Separated composite audio examples' channel,sample rate,format parameters from commom macro to in dedicated macro and out dedicated macro.
 - * replaced USB_PrepareData with USB_AudioRecorderGetBuffer.

- 2.4.1
 - New features:
 - * Added enumeration fail callback to host stack when the attached device's enumeration failed.
- 2.4.0
 - Improvement:
 - * Device Charger Detection (DCD) software architecture was refactored.
 - New features:
 - * Enabled Device Charger Detection (DCD) on RT1060.
 - * Enabled Device Charger Detection on RT600.
 - * Enabled host battery charger function on RT600.
- 2.3.0
 - New features:
 - * Added host video camera support. example: `usb_host_video_camera`
 - * Added a new device example. example: `usb_device_composite_cdc_hid_audio_unified`
- 2.2.0
 - New features:
 - * Added device DFU support.
 - * Supported OM13790DOCK on LPCXpresso54018.
 - * Added multiple logical unit support in msc class driver, updated `usb_device_lba_information_struct_t` to support this.
 - * Supported multiple transfers for host ISO on IP3516HS.
 - Bug fixes:
 - * Fixed device ip3511 prime data length than maxpacket size issue.
 - * Initialized interval attribute in `usb_device_endpoint_struct_t/usb_device_endpoint_init_struct_t`.
 - * Removed unnecessary header file in device CDC class driver, removed unnecessary `usb_echo`, and added `DEBUG` macro for necessary `usb_echo` in device CDC class driver.
 - * Fixed device IP3511HS unfinished interrupt transfer missing issue.
- 2.1.0
 - New features:
 - * Added host RNDIS support. example: `lwip_dhcp_usb`
 - * Enabled USB 3.0 support on device stack.
 - * Power Delivery feature: Added OM13790HOST support; Added auto policy feature; Printed e-marked cable information;
- 2.0.1
 - Bug fixes:
 - * Fixed some USB issues: Fixed MSC CV test failed in MSC examples.
 - * Changed 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 Component Change Log

CODEC

The current codec common driver version is 2.3.1.

- 2.3.1
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 16.1,16.3.
- 2.3.0
 - Improvements
 - * Added enum `_codec_volume_capability` for `CODEC_SetVolume/CODEC_SetMute` to cover more volume configurations.
- 2.2.2
 - Bug Fixes
 - * Fixed the typo in codec common driver.
- 2.2.1
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.2.0
 - Improvements
 - * Used `HAL_CODEC_HANDLER_SIZE` which is determined by low level driver instead of use `CODEC_HANDLE_SIZE` for the codec device handle definition.
- 2.1.1
 - Improvements
 - * Supported all of the codec in the codec adapter.
 - * Modified the codec handle definition to improve user experience.
 - * Modified the capability member type from entity to pointer in codec handle.
 - Bug Fixes
 - * Fixed the Coverity issue regrading array compared against 0.
- 2.1.0
 - Deprecated APIs
 - * `CODEC_GetMappedFormatBits`
 - * `CODEC_I2C_WriteReg`
 - * `CODEC_I2C_ReadReg`
 - * `CODEC_I2C_ModifyReg`
 - * `CODEC_SetEncoding`
 - new APIs
 - * `CODEC_SetPower`
 - * `CODEC_SetVolume`
 - * `CODEC_SetMute`
 - * `CODEC_SetPlay`
 - * `CODEC_SetRecord`
 - * `CODEC_SetRecordChannel`

- * CODEC_ModuleControl
- new features
 - * Removed duplicate members in codec_handle_t and codec_config_t.
 - * Added codec_config_t pointer in codec_handle_t.
 - * Added codec capability flag in codec_handle_t.
 - * Used codec adapter instead of function pointer in codec common driver.
- 2.0.1
 - Added delayMs function pointer in codec handle.
- 2.0.0
 - Initial version.

WM8904

The current wm8904 driver version is 2.5.1.

- 2.5.1
 - Bug Fixes
 - * Fixed invalid clock divider issue generated from WM8904_SetMasterClock api
 - * Replace ‘__REV16’ with general implementation to swap bytes in a short variable.
- 2.5.0
 - Improvements
 - * Added master clock configuration support in function WM8904_SetAudioFormat.
 - * Align the sysclk paramter definition for the WM8904_SetAudioFormat/WM8904_SetMasterClock.
 - * Added api WM8904_SetDACVolume to support adjust DAC volume.
 - * Fixed the MISRA-2012 violation of 12.2, 10.3.
- 2.4.4
 - Bug Fixes
 - * Added the 11.025kHz/22.05kHz/44.1kHz samplerate support on codec WM8904.
 - * Fixed the MISRA-2012 violation of 4.7.
- 2.4.3
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.6, 9.3, 10.1, 10.3, 10.4, 10.7, 10.8, 11.8, 11.9, 14.4, 16.1, 16.3, 16.4, 17.7, 20.9.
- 2.4.2
 - Bug Fixes
 - * Corrected the volume setting function behavior in wm8904 driver, support range align with its specification range.
 - * Corrected the volume setting function behavior in wm8904 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.4.1
 - Bug Fixes
 - * Fixed the bit width register field overwritten issue.

- 2.4.0
 - New features
 - * Added flt support in wm8904 driver.
- 2.3.0
 - Improvements
 - * Added new API WM8904_SetMasterClock to support BCLK/LRCLK output mode.
- 2.1.0
 - new APIs
 - * WM8904_ReadRegister
 - * WM8904_WriteRegister
 - * WM8904_ModifyRegister
 - * WM8904_SetRecord
 - * WM8904_SetPlay
 - * WM8904_SetRecordChannel
 - * WM8904_SetModulePower
 - * WM8904_SetChannelVolume
 - * WM8904_SetChannelMute

New features

- Removed dependency on codec common driver.
- Added dependency on codec i2c.

Bug Fixes

- Fixed unchecked return value in WM8904_Deinit.
- Fixed the alignment fault issue by adding __NOP between continuous memory access.

2.0.3

- Bug Fixes
 - Fixed issue that wm8904 register access function truncated return value.

2.0.2

- Bug Fixes
 - Fixed using uninitialized value format.fsRatio when calling WM8904_UpdateFormat.

2.0.1

- Added WM8904_CheckAudioFormat API.
- Changed the second parameter's name of WM8904_SetAudioFormat to sysclk.

2.0.0

- Initial version.

.1 WM8960

The current wm8960 driver version is 2.2.1.

- 2.2.1
 - Bug fixes
 - * Improved the internal PLL fatctor calculation formula.
- 2.2.0
 - Improvements
 - * Added masterClock member in wm8960_config_t to support wm8960 master mode.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 4.7, 5.8, 10.3, 10.4, 12.2, 14.4.
 - * Added the bit clock divider configuration when wm8960 act as master.
- 2.1.3
 - Bug Fixes
 - * Fixed the issue that WM8960 had no ack when performing write register by updating the byte count to be written.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.1.2
 - Improvements
 - * Enabled the class D output in WM8960_Init.
 - Bug Fixes
 - * Corrected the volume setting function behavior in wm8960 driver, support range aligned with its specification range.
 - * Corrected the volume setting function behavior in wm8960 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.1.1
 - Improvements
 - * Removed useless bit clock divider configuration in function WM8960_ConfigDataFormat.
- 2.1.0
 - Improvements
 - * Added new API WM8960_SetPlay.
 - * Fixed error status overwrite issue in WM8960_ConfigDataFormat function.
 - * Removed dependency on codec common driver.
 - * Added dependency on codec i2c.
 - Bug Fixes
 - * Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.2
 - Removed bit width hard code setting in function WM8960_SetProtocol.
- 2.0.1
 - Corrected the bclk divider calculation.
- 2.0.0
 - Initial version.

SGTL5000

The current sgtl5000 driver version is 2.1.1.

- 2.1.1
 - Improvements
 - * Corrected the volume setting function behavior in SGTL5000 driver, support range align with its specification range.
 - * Corrected the volume setting function behavior in SGTL5000 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.3, 8.3, 10.7, 17.7.
- 2.1.0
 - Improvements
 - * Added API SGTL_SetPlay/SGTL_SetRecord.
 - * Removed dependency on codec common driver.
 - * Added dependency on codec i2c.
 - * Fixed division or modulo by zero issue in SGTL_ConfigDataFormat function.
 - Bug Fixes
 - * Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.0
 - Initial version.

DA7212

The current da7212 driver version is 2.2.3.

- 2.2.3
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 10.4, 4.7.
- 2.2.2
 - Bug Fixes
 - * Fixed the MISRA-2012 violations.
 - Fixed rule 8.6, 9.3, 10.1, 10.3, 10.4, 10.7, 10.9, 11.1, 11.8, 14.4, 16.1, 16.3, 17.7, 17.3, 17.7, 20.9.
- 2.2.1
 - Improvements
 - * Corrected the volume setting function behavior in DA7212 driver, support range align with its specification range.
 - * Corrected the volume setting function behavior in DA7212 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
- 2.2.0
 - Improvements
 - * Added bclk invert parameter in the format structure.
 - * Added API DA7212_SetMasterModeBits/DA7212_SetPLLConfig.
 - * Added pll/sysClkSource parameters in the da7212 configuration structure.
 - * Disabled PLL by default.
- 2.1.0

- Improvements
 - * Removed dependency on codec common driver.
 - * Added dependency on codec i2c.
- Bug Fixes
 - * Fixed the alignment fault issue by adding __NOP between continuous memory access.
- 2.0.0
 - Initial version.

CS42888

The current cs42888 driver version is 2.1.3

- 2.1.3
 - Improvements
 - * Removed the assertion for codec reset function pointer.
- 2.1.2
 - Improvements
 - * Corrected the volume setting function behavior in CS42888 adapter, support range 0 - 100, 0 for mute, 100 for maximum volume.
 - Bug Fixes
 - * Fixed violations of MISRA C-2012 rule 4.7, 10.3, 8.3, 10.7, 17.7.
 - * Corrected the channel index during setting AIN volume in CS42888_Init.
- 2.1.1
 - Improvements
 - * Used software delay with delayMs pointer not provided by application.
 - * Fixed error status overwrite issue in CS42888_Init function.
 - * Removed dependency on codec common driver.
 - * Added API CS42888_SelectFunctionalMode/CS42888_SetChannelMute.
 - * Added dependency on codec i2c.
- 2.1.0
 - Improvements
 - * Unified CS42888 codec driver interface.
 - * Bug Fixes
 - Corrected the ADC/DAC functional mode macro definition.
 - Added TDM and OLM mode support in the function CS42888_SetProtocol.
- 2.0.0
 - Initial version.

SERIAL_MANAGER

The current Serial_Manager component version is 1.0.2.

- 1.0.2
 - Add SerialManager_WriteTimeDelay()/SerialManager_ReadTimeDelay() for serial manager's

- read/write non-blocking mode.
- 1.0.1
 - Add prefixing fsl_component_XXX/fsl_adapter_XXX.
- 1.0.0
 - 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.

