MCUXpresso SDK Release Notes Supporting evkmimx8mn

Change Logs

Contents

Driver Change Log

CLOCK

The current CLOCK driver version is 2.0.0

- 2.0.0
  - Initial version.

ECSPI

The current eCSPI driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed violations of MISRA C-2012 rules: 10.1, 10.3, 10.4
- 2.0.1
  - Bug Fixes
    - Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
    - Fixed sign-compare warning in ECSPI_SendTransfer.
- 2.0.0
  - Initial version.

GPT

The current GPT driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.1, 10.3, 10.4, 10.6, 10.8, 17.7.
- 2.0.0
  - Initial version.

GPIO

The current GPIO driver version is 2.0.3.

- 2.0.3 -Bug Fixes
  - MISRA C-2012 issue fixed.
    - Fixed rules, containing: rule-10.3, rule-14.4, and rule-15.5.
- 2.0.2
  - Bug Fixes

        * Fixed the bug of enabling wrong GPIO clock gate in initial API. Since some GPIO instances may not have a clock gate enabled, it checks the clock gate number and makes sure the clock gate is valid.
- 2.0.1
  - Improvements
    * API interface changes:
      · Refined naming of the API while keeping all original APIs, marking them as deprecated. Original APIs will be removed in next release. The main change is to update the API with prefix of _PinXXX() and _PortXXX().
- 2.0.0
  - Initial version.

I2C

The current I2C driver version is 2.0.5.

- 2.0.5
  - Bug Fixes
    * Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
    * Fixed MISRA issues.
      · Fixed rules 10.1, 10.3, 10.4, 11.9, 14.4, 15.7, 16.4, 17.7.
  - Improvements
    * Updated the I2C_WAIT_TIMEOUT macro to unified name I2C_RETRY_TIMES.
- 2.0.4
  - Bug Fixes
    * Fixed the issue that I2C Master transfer APIs(blocking/non-blocking) did not support the situation that master transfer with subaddress and transfer data size being zero, which means no data followed by the subaddress.
- 2.0.3
  - Improvements
    * Improved code readability, added new static API I2C_WaitForStatusReady for the status flag wait, and changed to call I2C_WaitForStatusReady instead of polling flags with reading register.
- 2.0.2
  - Improvements
    * Added I2C_WATI_TIMEOUT macro to allow users to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.1
  - Bug Fixes
    * Added a proper handle for transfer config flag kI2C_TransferNoStartFlag to support transmit with kI2C_TransferNoStartFlag flag. Only supports write only or write+read with no start flag; does not support read only with no start flag.
- 2.0.0
  - Initial version.

TMU

The current TMU driver version is 2.0.2.

- 2.0.2
  - Other Changes:
    - Removed the gain setting and reference voltage setting of amplifier.
- 2.0.1
  - Bug Fixes
    - Fixed missing right pair definition for extern C.
- 2.0.0
  - Initial version.
  - This module was first developed on i.MX 8MM.

PDM

The current PDM driver version is 2.2.1.

- 2.2.1
  - Bug Fixes
    - Fixed violation of MISRA C-2012 Rule 10.1, 10.3, 10.4, 10.6, 10.7, 11.3, 11.8, 14.4, 17.7, 18.4.
    - Added medium quality mode support in function PDM_SetSampleRateConfig.
- 2.2.0
  - Improvements
    - Added api PDM_SetSampleRateConfig to improve user experience and marked api PDM_SetSampleRate as deprecated.
- 2.1.1
  - Improvements
    - Used new SDMA API SDMA_SetDoneConfig instead of SDMA_EnableSwDone for PDM SDMA driver.
- 2.1.0
  - Improvements
    - Added software buffer queue for transactional API.
- 2.0.1
  - Improvements
    - Improved HWVAD feature.
- 2.0.0
  - Initial version.

PWM

The current PWM driver version is 2.0.1.

- 2.0.1

  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 17.7.
- 2.0.0
  - Initial version.

UART

The current UART driver version is 2.0.2.

- 2.0.2
  - Improvements
    - Added check for transmission complete in UART_WriteBlocking, UART_TransferHandleIRQ and UART_SendSDMACallback to ensure all the data would be sent out to bus.
    - Modified UART_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.
  - Bug Fixes
    - Fixed MISRA issues.
      - Fixed rules 10.1, 10.3, 10.4, 10.6, 10.7, 10.8, 11.9, 14.4.
- 2.0.1
  - Bug Fixes
    - Memset local variable SDMA transfer configuration structure to make sure unused members in structure are cleared.
- 2.0.0
  - Initial version.

MU

The Current MU driver version is 2.0.5.

- 2.0.5 -Bug Fixes
  - Fixed violations of the MISRA C-2012 rules 14.4, 15.5.
- 2.0.4
  - Improvements:
    - Improved for the platforms which don't support reset assert interrupt and get the other core power mode.
- 2.0.3 -Bug fix
  - MISRA C-2012 issue fixed.
    - Fix rules, containing: rule-10.3, rule-14.4, rule-15.5.
- 2.0.2
  - Added support for MIMX8MQx.
- 2.0.1
  - Added support for MCIMX7Ux_M4.
- 2.0.0

  - Initial version.

RDC

The current RDC driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 11.3, 11.8, 17.7.
- 2.0.1
  - Bug Fixes:
    - Added __DSB after new configuration is set to ensure the new configuration takes effect.
- 2.0.0
  - Initial version.

RDC_SEMA42

The current RDC_SEMA42 driver version is 2.0.2.

- 2.0.2
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 10.8, 14.3, 14.4, 18.1.
- 2.0.1
  - Improvements:
    - Added support for the platforms that don't have dedicated RDC_SEMA42 clock gate.
- 2.0.0
  - Initial version.

SAI

The current SAI driver version is 2.2.2.

-2.2.2

- Bug Fixes
  - Fixed the issue of MISRA 2004 rule 9.3.
  - Fixed sign-compare warning.
  - Fixed the PA082 build warning.
  - Fixed sign-compare warning.
  - Fixed violations of MISRA C-2012 rule 10.3,17.7,10.4,8.4,10.7,10.8,14.4,17.7,11.6,10.1,10.-6,8.4,14.3,16.4,18.4.
  - Allow to reset Rx or Tx FIFO pointers only when Rx or Tx is disabled.
- Improvements
  - Added 24bit raw audio data width support in sai sdma driver.

  - Disabled the interrupt/DMA request in the SAI_Init to avoid generates unexpected sai FIFO requests.

2.2.1

- Improvements
  - Added mclk post divider support in function SAI_SetMasterClockDivider.
  - Removed useless configuration code in SAI_RxSetSerialDataConfig.
- Bug Fixes
  - Fixed the SAI SDMA driver build issue caused by the wrong structure member name used in the function SAI_TransferRxSetConfigSDMA/SAI_TransferTxSetConfigSDMA.
  - Fixed BAD BIT SHIFT OPERATION issue caused by the FSL_FEATURE_SAI_CHANNEL_COUNTn.
  - Applied ERR05144: not set FCONT = 1 when TMR $>$ 0, otherwise the TX may not work.

2.2.0

- Improvements
  - Added new APIs for parameters collection and simplified user interfaces: SAI_Init SAI_SetMasterClockConfig SAI_TxSetBitClockRate SAI_TxSetSerialDataConfig SAI_TxSetFrameSyncConfig SAI_TxSetFifoConfig SAI_TxSetBitclockConfig SAI_TxSetConfig SAI_TxSetTransferConfig SAI_RxSetBitClockRate SAI_RxSetSerialDataConfig SAI_RxSetFrameSyncConfig SAI_RxSetFifoConfig SAI_RxSetBitclockConfig SAI_RXSetConfig SAI_RxSetTransferConfig SAI_GetClassicI2SConfig SAI_GetLeftJustifiedConfig SAI_GetRightJustifiedConfig SAI_GetTDMConfig

2.1.9

- Improvements
  - Improved SAI driver comment for clock polarity.
  - Added enumeration for SAI for sample inputs on different edges.
  - Changed FSL_FEATURE_SAI_CHANNEL_COUNT to FSL_FEATURE_SAI_CHANNEL_COUNTn(base) for the difference between the different SAI instances.
  - Added new APIs: SAI_TxSetBitClockDirection SAI_RxSetBitClockDirection SAI_RxSetFrameSyncDirection SAI_TxSetFrameSyncDirection

2.1.8

- Improvements
  - Added feature macro test for the sync mode2 and mode 3.
  - Added feature macro test for masterClockHz in sai_transfer_format_t.

2.1.7

- Improvements
  - Added feature macro test for the mclkSource member in sai_config_t.
  - Changed "FSL_FEATURE_SAI5_SAI6_SHARE_IRQ" to "FSL_FEATURE_SAI_SAI5_SAI6_SHARE_IRQ".
  - Added #ifndef #endif check for SAI_XFER_QUEUE_SIZE to allow redefinition.
- Bug Fixes

  - Fixed build error caused by feature macro test for mclkSource.

-2.1.6

- Improvements
  - Added feature macro test for mclkSourceClockHz check.
  - Added bit clock source name for general devices.
- Bug Fixes
  - Fixed incorrect channel numbers setting while calling RX/TX set format together.

-2.1.5

- Bug Fixes
  - Corrected SAI3 driver IRQ handler name.
  - Added I2S4/5/6 IRQ handler.
  - Added base in handler structure to support different instances sharing one IRQ number.
- New Features
  - Updated SAI driver for MCR bit MICS.
  - Added 192 KHZ/384 KHZ in the sample rate enumeration.
  - Added multi FIFO interrupt/SDMA transfer support for TX/RX.
  - Added an API to read/write multi FIFO data in a blocking method.
  - Added bclk bypass support when bclk is same with mclk.

2.1.4

- New Features
  - Added an API to enable/disable auto FIFO error recovery in platforms that support this feature.
  - Added an API to set data packing feature in platforms which support this feature.

2.1.3

- New Features
  - Added feature to make I2S frame sync length configurable according to bitWidth.

2.1.2

- Bug Fixes
  - Added 24-bit support for SAI eDMA transfer. All data shall be 32 bits for send/receive, as eDMA cannot directly handle 3-Byte transfer.

2.1.1

- Improvements
  - Reduced code size while not using transactional API.

2.1.0

- Improvements
  - API name changes:
    * SAI_GetSendRemainingBytes -> SAI_GetSentCount.
    * SAI_GetReceiveRemainingBytes -> SAI_GetReceivedCount.
    * All names of transactional APIs were added with "Transfer" prefix.
    * All transactional APIs use base and handle as input parameter.

        * Unified the parameter names.
- Bug Fixes
  - Fixed WLC bug while reading TCSR/RCSR registers.
  - Fixed MOE enable flow issue. Moved MOE enable after MICS settings in SAI_TxInit/SAI_-RxInit.

2.0.0

- Initial version.

SDMA

The current SDMA driver version is 2.3.0.

- 2.3.0
  - Improvements
    * Added peripheral-to-peripheral support in SDMA driver.
    * Added 24bit data width support in sdma driver.
  - Bug Fixes
    * Fixed Coverity issue: left shift may overflow issue.
    * Fixed MISRA2004 issue: the operand of underlying type 'unsigned char' or 'unsigned short' caused the result cast to the underlying type.
    * Fixed violations of MISRA C-2012 rule 10.3, 11.9, 10.4, 17.7, 20.7, 14.4, 11.6, 12.2, 16.4.
- 2.2.1
  - Bug Fixes
    * Fixed MISRA 2004 issue in sdma driver.
- 2.2.0
  - Improvements
    * Added fsl_sdma_script.h to define the sdma script address and firmware.
    * Updated the format of generic register R7 to align with newest firmware.
- 2.1.1
  - Improvements
    * Added SDMA_SetDoneConfig to support hardware/software done configuration.
    * Marked SDMC_EnableSwDone as deprecated.
  - Bug Fixes
    * Fixed logical dead code issue in function SDMA_SetDoneConfig.
- 2.1.0
  - Improvements
    * Added SDMA_SetMultiFifoConfig API to support multi fifo feature.
    * Added SDMA_EnableSwDone API to support software done feature.
    * Added SDMA_LoadScript API to support load script to SDMA program memory.
    * Added SDMA_DumpScript API to support dump script from SDMA program memory.
    * Added SDMA3 IRQ handler.
- 2.0.0
  - Initial version.

SEMA4

The current SEMA4 driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed violations of the MISRA C-2012 rules 10.3, 10.4, 15.5, 18.1, 18.4.
- 2.0.0
  - Initial version.

WDOG

The current WDOG driver version is 2.1.1.

- 2.1.1
  - Bug Fixes
    - MISRA C-2012 issue fixed: rule 10.1, 10.3, 10.4, 10.6, 10.7 and 11.9.
    - Fixed the issue of the inseparable process interrupted by other interrupt source.
      - WDOG_Init
      - WDOG_Refresh
- 2.1.0
  - New Features
    - Added new API "WDOG_TriggerSystemSoftwareReset()" to allow users to reset the system by software.
    - Added new API "WDOG_TriggerSoftwareSignal()" to allow users to trigger a WDOG_B signal by software.
    - Removed the parameter "softwareAssertion" and "softwareResetSignal" out of the wdog_config_t structure.
    - Added new parameter "enableTimeOutAssert" to the wdog_config_t structure. With this parameter enabled, when the WDOG timeout occurs, a WDOG_B signal will be asserted. This signal can be routed to external pin of the chip. Note that WDOG_B signal remains asserted until a power-on reset (POR) occurs.
- 2.0.1
  - New Features
    - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
  - Initial version.

ASRC

The current ASRC driver version is 2.0.1.

- 2.0.1
  - Bug Fixes
    - Fixed the context id hard code issue in the function ASRC_TransferInCreateHandleSD-

MA/ASRC_TransferOutCreateHandleSDMA.
  - Improvements
    * Added support for the data size bigger than 64K in sdma driver.
- 2.0.0
  - Initial version.

RTOS Change Log

FreeRTOS for MCUXpresso SDK.

The current version is Amazon-FreeRTOS 201908.00 Original package is available at github.-com/aws/amazon-freertos.

- 201908.00_rev0
  - update amazon freertos version
  - Fix freertos_tasks_c_additions.h - fix IAR build fail
  - update queue.c - add definition for pvBuffer necessary for segger sysview
  - iot_crypto.c - change include file to be possible include mbedtls config file defined by MBEDTLS_CONFIG_FILE macro
  - iot_mqtt_agent.h - extend MQTTAgentConnectParams_t structure - required by se_hostlib examples
  - Fixed build warnings:
    - aws_dev_mode_key_provisioning.c - some variables were declared but never referenced
    - aws_iot_network_manager.c - some functions were declared but never referenced
    - iot_device_metrics.c - add include
    - iot_pkcs11_mbedtls.c - incompatible pointer type, unused variable
    - iot_demo_freertos.c - macro expansion producing 'defined' has undefined behavior
    - iot_pkcs11_mbedtls.c - comparison of address not equal to a null pointer is always true
    - pkcs11.h - '__PASTE' macro redefined
- 1.4.9_rev0
  - Remove 3rd party libraries lwip, mbedtls (use MCUXpresso SDK versions).
  - Add missing comments to heap_useNewlib.c.
- 1.4.7_rev0
  - New features:
    - Add optional allocation scheme heap_useNewlib.c by D. Nadler.
    - Enable task aware debugging for cm33 platforms
    - Move tickless implementation to application layer
  - Other changes:
    - Fix other build warnings, errors
- 1.4.6_rev0
  - New features:
    - Update support of CM33 port with Trustzone, MPU, FPU support
    - Add support for AWS test for Cypress WiFi
    - Use lwip netif api to avoid lwIP raw API calls outside of tcpip thread in aws_wifi.c
  - Other changes:
    - Fix issues with mflash driver
    - Fix other build warnings, errors
- 1.4.0_rev1
  - New features:
    - Add implementation of vTaskEndScheduler for CM0 GCC port.

        * Support for CM33, CM33F architectures based on CM3, CM4F ports
- 1.4.0_rev0
  - New features:
    * Support for pkcs11 for several platforms, secure element host library under pkcs11/portable/nxp folder
    * Lwip, wifi_qca support for secure_sockets in secure_sockets/portable/nxp folder
    * Flash driver support for several platforms in third_party/mcu_vendor/nxp folder
    * Generic support for aws_wifi under wifi/portable/nxp/common folder
  - Other changes:
    * Fix several build warnings, errors

Updates applied to FreeRTOS kernel up to version 10.0.0 (up to Amazon - FreeRTOS merge). New kernel related changes will be described in section above as part of AWS package.

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

    * Tickless idle mode support.
    * Added template application for Kinetis Expert (KEx) tool (template_application).
- Other changes:
    * Folder structure reduction. Keep only Kinetis related parts.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

NXP 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.