
MCUXpresso SDK Release Notes Supporting LPC5410x

Change Logs

Contents

Driver Change Log	1
ADC	1
CLOCK	1
POWER	2
RESET	2
SPI_CMSIS	3
USART	3
COMMON	3
CRC	4
CTIMER	4
DMA	4
FLASHIAP	5
FMEAS	6
GINT	6
GPIO	6
MAILBOX	7
MRT	7
I2C	7
INPUTMUX	8
IOCON	8
PINT	8

Contents

Title	Page Number
RIT	9
RTC	10
SCTIMER	10
SYSCON	10
UTICK	10
SPI	11
USART	11
WWDT	12
IAP	12
Middleware Change Log	14
FatFs for MCUXpresso SDK	14
emWin library	14
LittlevGL for KSDK	14
Multicore SDK	14
RTOS Change Log	17
FreeRTOS for MCUXpresso SDK.	17

1 Driver Change Log

ADC

The current ADC driver version is 2.3.2.

- 2.3.2
 - Improvements
 - * Added delay after enabling using the ADC GPADC_CTRL0 LDO_POWER_EN bit for JN5189/QN9090.
- 2.3.1
 - Bug Fixes
 - * Avoided writing ADC STARTUP register in ADC_Init().
 - * Fixed Coverity zero divider error in ADC_DoSelfCalibration().
- 2.3.0
 - Improvements
 - * Updated "ADC_Init()" "ADC_GetChannelConversionResult()" API and "adc_resolution_t" structure to match QN9090.
 - * Added "ADC_EnableTemperatureSensor" API.
- 2.2.1
 - Improvements
 - * Added a brief delay in uSec after ADC calibration start.
- 2.2.0
 - Improvements
 - * Updated "ADC_DoSelfCalibration" API and "adc_config_t" structure to match LPC845.
- 2.1.0
 - Improvements
 - * Renamed "ADC_EnableShresholdCompareInterrupt" to "ADC_EnableThresholdCompareInterrupt".
- 2.0.0
 - Initial version.

CLOCK

The current CLOCK driver version is 2.3.0.

- 2.3.0
 - Replace the delay function
- 2.2.0
 - Optimization:
 - * remove some peripheral clock parameters in CLOCK_GetFreq function
- 2.1.0
 - New feature

- Adding new API `CLOCK_DelayAtLeastUs()` implemented by DWT to allow users set delay in unit of microsecond.

2.0.5

- Bug Fix:
 - Fix C++ build errors in `CLOCK_GetClockAttachId()` and `CLOCK_AttachClk()`.

2.0.4

- Bug Fix:
 - Fix attach incorrect `attach_id`.

2.0.3

- New Feature:
 - add get actual clock attach id api to allow users to obtain the actual clock source in target register.
- Bug Fix:
 - The attach clock and get actual clock attach id apis should check combination of two clock source.
 - Set selector of `CM_CLKOUTCLKSELB` to 3.
- Optimization:
 - Make the judgement statments more clear.
 - Strengthen the compatibility of clock attach id.
 - Remove some unmeaningful definitions and add some useful ones to enhance readability.

2.0.2

- some minor fixes.

2.0.0

- initial version.

POWER

The current POWER driver version is 2.0.0.

- 2.0.0
 - initial version.

RESET

The current RESET driver version is 2.0.1.

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

SPI_CMSIS

Current spi_cmsis driver version is 2.0

- 2.0
 - - Initial version.

USART

Current USART CMSIS driver version is 2.0

- 2.0
 - Initial version.

COMMON

The current COMMON driver version is 2.1.3.

- 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.

CRC

The current CRC driver version is 2.0.1.

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

CTIMER

The current CTimer driver version is 2.0.2.

- 2.0.2
 - New Features
 - * Added new API "CTIMER_GetTimerCountValue" to get the current timer count value.
 - * Added a control macro to enable/disable the RESET and CLOCK code in current driver.
 - * Added a new feature macro to update the API of CTimer driver for lpc8n04.
- 2.0.1
 - Improvements
 - * API Interface Change
 - Changed API interface by adding CTIMER_SetupPwmPeriod API and CTIMER_UpdatePwmPulsePeriod API, which both can set up the right PWM with high resolution.
- 2.0.0
 - Initial version.

DMA

The current DMA driver version is 2.4.0.

- 2.4.0
 - Improvements:
 - * Added new apis DMA_LoadChannelDescriptor/DMA_ChannelIsBusy to support polling transfer case.
 - Bug fix:
 - * Add address alignment check for descriptor source and destination address.
 - * Add DMA_ALLOCATE_DATA_TRANSFER_BUFFER for application buffer allocation.
- 2.3.0
 - Bug fix:

- * Removed DMA_HandleIRQ prototype definition from header file.
- * Added DMA_IRQHandle prototype definition in header file.
- 2.2.5
 - Improvements:
 - * Added new API DMA_SetupChannelDescriptor to support configure wrap descriptor.
 - * Added wrap support in function DMA_SubmitChannelTransfer.
- 2.2.4
 - Bug fix:
 - * Fixed the macro DMA_CHANNEL_CFER use wrong parameter to calculate DSTINC issue.
- 2.2.3
 - Bug fix:
 - * Improved DMA driver Deinit function for correct logic order.
 - Improvement:
 - * Added API DMA_SubmitChannelTransferParameter to support create head descriptor directly.
 - * Added API DMA_SubmitChannelDescriptor to support ping pong transfer.
 - * Added macro DMA_ALLOCATE_HEAD_DESCRIPTOR/DMA_ALLOCATE_LINK_DESCRIPTOR to simplify DMA descriptor allocation.
- 2.2.2
 - Bug fix:
 - * Do not use software trigger when hardware trigger is enabled.
- 2.2.1
 - Bug fix:
 - * Fixed coverity issue.
- 2.2.0
 - Improvements:
 - * Changed API DMA_SetupDMADescriptor to non-static.
 - * Marked below API as deprecated. DMA_PrepareTransfer. DMA_Submit transfer.
 - * Added below new API: DMA_SetChannelConfig. DMA_PrepareChannelTransfer. DMA_A_InstallDescriptorMemory. DMA_SubmitChannelTransfer. DMA_SetChannelConfigValid. DMA_DoChannelSoftwareTrigger. DMA_LoadChannelTransferConfig.
- 2.0.1
 - Improvement:
 - * Added volatile for DMA descriptor member xfercfg to avoid optimization.
- 2.0.0
 - Initial version.

FLASHIAP

The current FLASHIAP driver version is 2.0.3.

- 2.0.3
 - The FLASHIAP driver is marked as deprecated and will be removed in next release. All of

its APIs are moved to the IAP driver. The names of FLASHIAP's APIs are updated from FLASHIAP_XXX() to IAP_XXX().

- 2.0.2
 - Added the API for extended flash signature
- 2.0.1
 - Removed two incorrect commands.
- 2.0.0
 - Initial version.

FMEAS

The current FMEAS driver version is 2.1.0.

- 2.1.0
 - Updated "FMEAS_GetFrequency", "FMEAS_StartMeasure", "FMEAS_IsMeasureComplete" API and add definition to match ASYNC_SYSCON.
- 2.0.0
 - Initial version ported from LPCOpen.

GINT

The current GINT driver version is 2.0.1.

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

GPIO

The current GPIO driver version is 2.1.3.

- 2.1.4
 - Improvements
 - * Added API GPIO_PortGetInterruptStatus to retrieve interrupt status for whole port.
 - * Corrected typos in header file.
- 2.1.3
 - Improvements
 - * Updated "GPIO_PinInit" API. If it has DIRCLR and DIRSET registers, use them at set 1 or clean 0.
- 2.1.2
 - Improvements
 - * Removed deprecated APIs.
- 2.1.1

- Improvements
 - * API interface changes:
 - Refined naming of APIs while keeping all original APIs, marking them as deprecated. Original APIs will be removed in next release. The mainin change is updating APIs with prefix of `_PinXXX()` and `_PorortXXX`
- 2.1.0
 - New Features
 - * Added GPIO initialize API.
- 2.0.0
 - Initial version.

MAILBOX

The current MAILBOX driver version is 2.1.0.

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

MRT

The current MRT driver version is 2.0.1.

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

I2C

The current I2C driver version is 2.0.4.

- 2.0.4
 - Bug Fixes
 - * Fixed wrong assignment for `datasize` in `I2C_InitTransferStateMachineDMA`.
 - * Fixed wrong working flow in `I2C_RunTransferStateMachineDMA` to ensure master can work in no start flag and no stop flag mode.
 - * Fixed wrong working flow in `I2C_RunTransferStateMachine` and added `kReceiveData-BEGINState` in `_i2c_transfer_states` to ensure master can work in no start flag and no stop flag mode.
 - * Fixed wrong handle state in `I2C_MasterTransferDMAHandleIRQ`. After all the data has been transfered or nak is returned, handle state should be changed to idle.

- Improvements
 - * Rounded up the calculated divider value in I2C_MasterSetBaudRate.
- 2.0.3
 - Bug Fixes
 - * Fixed Coverity issue of unchecked return value in I2C_RTOS_Transfer.
- 2.0.2
 - New Features
 - * Added macro gate "FSL_SDK_ENABLE_I2C_DRIVER_TRANSACTIONAL_APIS" to enable/disable the transactional APIs which will help reduce the code size when no non-blocking transfer is used. Default configuration is enabled.
 - * Added a control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
 - Improvements
 - * Added I2C_WATI_TIMEOUT macro to allow the user to specify the timeout times for waiting flags in functional API and blocking transfer API.
- 2.0.0
 - Initial version.

INPUTMUX

The current INPUTMUX driver version is 2.0.1.

- 2.0.1
 - Support channel mux setting in INPUTMUX_EnableSignal().
- 2.0.0
 - Initial version.

IOCON

The current IOCON driver version is 2.1.1.

- 2.1.1
 - Updated left shift format with mask value instead of a constant value to automatically adapt to all platforms.
- 2.1.0
 - Added a new IOCON_PinMuxSet() function with a feature IOCON_ONE_DIMENSION for LPC845MAX board.
- 2.0.0
 - Initial version.

PINT

The current PINT driver version is 2.1.4.

- 2.1.4
 - Improvements
 - * Added feature to control distinguish PINT/SECPINT relevant interrupt/clock configurations for PINT_Init and PINT_Deinit API.
 - * Swapped the order of clearing PIN interrupt status flag and clearing pending NVIC interrupt in PINT_EnableCallback and PINT_EnableCallbackByIndex function.
 - Bug Fixes
 - * Fixed build issue caused by incorrect macro definitions.
- 2.1.3
 - Bug fix:
 - * Updated PINT_PinInterruptClrStatus to clear PINT interrupt status when the bit is asserted and check whether was triggered by edge-sensitive mode.
 - * Write 1 to IST corresponding bit will clear interrupt status only in edge-sensitive mode and will switch the active level for this pin in level-sensitive mode.
 - * Fixed MISRA c-2012 rule 10.1, rule 10.6, rule 10.7.
 - * Added FSL_FEATURE_SECPINT_NUMBER_OF_CONNECTED_OUTPUTS to distinguish IRQ relevant array definitions for SECPINT/PINT on lpc55s69 board.
 - * Fixed PINT driver c++ build error and remove index offset operation.
- 2.1.2
 - Improvement:
 - * Improved way of initialization for SECPINT/PINT in PINT_Init API.
- 2.1.1
 - Improvement:
 - * Enabled secure pint interrupt and add secure interrupt handle.
- 2.1.0
 - Added PINT_EnableCallbackByIndex/PINT_DisableCallbackByIndex APIs to enable/disable callback by index.
- 2.0.2
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.1
 - Bug fix:
 - * Updated PINT driver to clear interrupt only in Edge sensitive.
- 2.0.0
 - Initial version.

RIT

The current RIT driver version is 2.1.0.

- 2.1.0
 - Fixed issue for wrong implementation of clearing counter API in RIT driver.
- 2.0.2
 - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.1

- Fixed incorrect comments of some APIs.
- 2.0.0
 - Initial version.

RTC

The current RTC driver version is 2.0.0.

- 2.0.0
 - Initial version.

SCTIMER

The current SCTimer driver version is 2.1.1.

- 2.1.1
 - Improvements
 - * Updated the register and macro names to align with the header of devices.
- 2.1.0
 - Bug Fixes
 - * Fixed issue where SCT application level Interrupt handler function is occupied by SCT driver.
 - * Fixed issue where wrong value for INSYNC field inside SCTIMER_Init function.
 - * Fixed issue to change Default value for INSYNC field inside SCTIMER_GetDefault-Config.
- 2.0.1
 - New Features
 - * Added control macro to enable/disable the RESET and CLOCK code in current driver.
- 2.0.0
 - Initial version.

SYSCON

The current SYSCON driver version is 2.0.0.

- 2.0.0
 - Initial version.

UTICK

The current UTICK driver version is 2.0.2.

- 2.0.2

- Added new feature definition macro to enable/disable power control in drivers for some devices have no power control function.
- 2.0.1
 - Added control macro to enable/disable the CLOCK code in current driver.
- 2.0.0
 - Initial version.

SPI

The current SPI driver version is 2.0.2.

- 2.0.2
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
 - Added wait mechanism in SPI_MasterTransferBlocking() API, check if master SPI become IDLE when the EOT bit is set before returning. This confirms all data has been sent out by SPI master.

2.0.1

- Optimized the code to raise the performance of blocking and nonblocking transfer.
 - Added new feature to support the ignore receive function, this update makes the transfer faster if no need to received data, both polling transfer and nonblocking transfr can support this feature.
 - Changed the default value of TX FIFO size and RX FIFO size in SPI_MasterGetDefaultConfig() and SPI_SlaveGetDefaultConfig() to 0, because different devices have different FIFO size, a fixed value may not very adaptable. So, users need to configure the TX and RX FIFO size in their applications as they want if FIFO is enabled.

2.0.0

- Initial version.

USART

The current USART driver version is 2.1.0.

- 2.1.0
 - New feature:
 - * Added new feature to allow users configure the USART to synchronous transfer(master and slave) mode.
- 2.0.1
 - Added control macro to enable/disable the RESET and CLOCK code in current driver.
 - Added new API to allow users enable the CTS which determines whether CTS is used for flow control.
- 2.0.0
 - Initial version.

WWDT

The current WWDT driver version is 2.1.3.

- 2.1.3
 - Bug Fixes
 - * Fixed legacy issue when initializing the MOD register.
- 2.1.2
 - Improvements
 - * Updated the "WWDT_ClearStatusFlags" API and "WWDT_GetStatusFlags" API to match QN9090. WDTOF is not set in case of WD reset. Get info from PMC instead.
- 2.1.1
 - New Features
 - * Added new feature definition macro for devices which have no LCOK control bit in MOD register.
 - * Implemented delay/retry in WWDT driver.
- 2.1.0
 - Improvements
 - * Added new parameter in configuration when initializing WWDT module. This parameter, which must be set, allows the user to deliver the WWDT clock frequency.
- 2.0.0
 - Initial version.

IAP

The current IAP driver version is 2.0.3.

- 2.0.3
 - New Features
 - * Added support for LPC 845's FAIM operation.
 - * Added support for LPC 80x's fixed reference clock for flash controller.
 - * Added support for LPC 5411x's Read UID command useless situation.
 - Improvements
 - * Improved the document and code structure.
- 2.0.2
 - New Features
 - * Added an API to read generated signature.
 - Bug Fixes
 - * Fixed the incorrect board support of IAP_ExtendedFlashSignatureRead().
- 2.0.1
 - New Features
 - * Added an API to read factory settings for some calibration registers.
 - Improvements
 - * Updated the size of result array in part APIs.
- 2.0.0

- Initial version.

2 Middleware Change Log

FatFs for MCUXpresso SDK

Current version is FatFs R0.13c_rev0.

- R0.13c_rev0
 - Upgraded to version 0.13c
 - Apply 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.

emWin library

The currently supported version is 5.48r.

LittlevGL for KSDK

- 5.3_rev1
 - Integrate LittlevGL 5.3 to SDK.

Multicore SDK

The current version of Multicore SDK is 2.6.0.

- 2.6.0
 - Multicore SDK component versions:

- * embedded Remote Procedure Call (eRPC) v1.7.2
- * eRPC generator (erpcgen) v.1.7.2
- * 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) v.1.7.1
 - * 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) v.1.7.0
 - * 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 `rpmsg_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 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 annotation for functions.
 - * eRPC: Added support

3 RTOS Change Log

FreeRTOS for MCUXpresso SDK.

The current version is Amazon-FreeRTOS 1.4.0 Original package is available at github.com/aws/amazon-freertos.

- 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 -fno 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 appropriate 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.

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.

© 2018 NXP B.V.