NXP Semiconductors
User's Guide

Document Number: MCUXSDKFRTOSGSUG
Rev. 5, 04/2019

Getting Started with MCUXpresso SDK and FreeRTOS OS

Contents

1 Overview....1
2 FreeRTOS OS role in MCUXpresso SDK....1
3 FreeRTOS source description....2
4 FreeRTOS package integration in MCUXpresso SDK....2
5 FreeRTOS drivers....2
6 FreeRTOS example applications....2
7 Revision history....4

1 Overview

The MCUXpresso Software Development Kit (MCUXpresso SDK) 2.5 provides a comprehensive software package with pre-integrated FreeRTOS OS. NXP provides FreeRTOS OS additions, such as RTOS driver wrappers, RTOS ready FatFs file system, and the implementation of FreeRTOS tickless mode. This document describes steps required to configure supported development tools used to build run, and debug applications with the FreeRTOS OS targeted for MCUXpresso SDK.

2 FreeRTOS OS role in MCUXpresso SDK

The MCUXpresso Software Development Kit (MCUXpresso SDK) is a software framework for developing applications on MCUs. The software components in the framework include peripheral drivers, middleware, and real-time operating systems.

This version of MCUXpresso SDK provides FreeRTOS OS version 10.0.1. Detailed information about the distribution can be found on www.freertos.org. The framework contains a set of examples which show basic FreeRTOS OS features. This makes it easy to start a new FreeRTOS project or begin

experimenting with FreeRTOS OS. Selected drivers and middleware are RTOS ready with related FreeRTOS adaptation layer.

3 FreeRTOS source description

Core files: Generic FreeRTOS core files are located in <SDK_DIR>/rtos/amazon-freertos/lib/FreeRTOS. Header files are in the <SDK_DIR>/rtos/amazon-freertos/lib/include subfolder.

Port files: FreeRTOS porting files are located in <SDK_DIR>/rtos/amazon-freertos/lib/FreeRTOS/portable.

Configuration files: Configuration files are designed as application specific and its location is dependent on the application location. Usually they are located in the example's root folder (for freertos_sem: <SDK_DIR>/boards/<board>/rtos_examples/freertos_sem). The main configuration file is FreeRTOSConfig.h.

4 FreeRTOS package integration in MCUXpresso SDK

- Removed files unrelated to SDK:
  - Removed extensions to the FreeRTOS OS (CLI, FAT_SL, and UDP).
  - Reduced FreeRTOS folder structure (remove nested folders).
- Added SystemCoreClock global variable to FreeRTOS port.c files.
- Enabled tickless mode.
- Enabled KDS Task Aware Debugger. Apply FreeRTOS patch to enable configRECORD_STACK_HIGH_ADDRESS macro.
- Enabled -flto optimization in GCC by adding __attribute__((used)) for vTaskSwitchContext.

5 FreeRTOS drivers

Selected drivers provide FreeRTOS support in the form of an additional layer. This solution enables simple driver integration in RTOS-based applications.

Drivers with FreeRTOS layers are:

- UART / LPUART / LPSCI
- I2C / LPI2C
- SPI / LPSPI / DSPI / ECSPI

The drivers for the FreeRTOS OS is a layer built on top of standard SDK peripheral drivers to achieve multithread (RTOS) awareness. The wrappers provide an API, which blocks the calling task until the I/O operation completes and allows other tasks to run in the background. This is achieved by using the asynchronous API of the underlying driver, along with RTOS task synchronization objects. Underlying drivers require enabled interrupts for proper operation.

It is recommended to use the FreeRTOS drivers instead of SDK bare metal drivers. The UART class drivers are demonstrated in the FreeRTOS UART examples. The example shows how to use UART class driver with the FreeRTOS OS for standard communication.

6 FreeRTOS example applications

The SDK provides a set of FreeRTOS OS-related applications. The examples are written to demonstrate basic FreeRTOS features and the interaction between peripheral drivers and the RTOS.

6.1 List of examples

Table 1. List of examples

FreeRTOS examples	Driver examples
freertos_hello	freertos_uart
freertos_event	freertos_lpuart
freertos_sem	freertos_lpsci
freertos_sem_static	freertos_i2c
freertos_mutex	freertos_lpi2c
freertos_queue	freertos_spi
freertos_swtimer	freertos_dspi
freertos_generic	freertos_ecspi
freertos_tickless	freertos_lpspi
freertos_percepio_snapshot	freertos_sdcard
freertos_segger_sysview	mmcccard_freertos
freertos_sdio	freertos_sdio

6.2 Building a FreeRTOS example application

The FreeRTOS OS in SDK is provided in the form of source files directly linked to the projects.

- <SDK_DIR> is the directory where the SDK package is installed on the hardware.
- <FREERTOS_DIR> is the directory where FreeRTOS OS is located within the SDK, specifically <SDK_DIR>/rtos/amazon-freertos.
- <board> replaces the name of the board (for example, TWR-K64F120M)
- <mcu> replaces the name of the processor (for example, MK64F120M)
- <tool> replaces the name of the toolchain (for example, IAR)
- <target> replaces the name of the project target (for example, Debug)

6.3 Building the first FreeRTOS application

- Open the workspace file with the related FreeRTOS example located in the <SDK_DIR>/boards/<board>/rtos_examples/<example>/<tool>/<example> folder.
- Build the required targets (for example, Debug) in all projects contained in the workspace.
- Run the application.

6.4 FreeRTOS aware debugger plugin

NXP provides FreeRTOS task aware debugger for GDB. The plugin is compatible with Eclipse-based tools (MCUXpresso) and can be obtained from the Eclipse install wizard (Help -> Install New Software...).

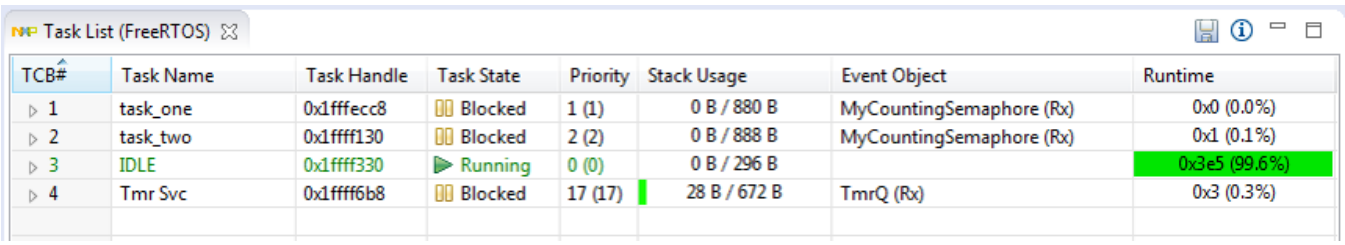

Figure 1. FreeRTOS TAD view

7 Revision history

This table summarizes revisions to this document.

Table 2. Revision history

Revision number	Date	Substantive changes
0	01/2016	Initial release
1	09/2016	Updated for LPC release
2	01/2017	Version update to 9.0.0
3	05/2018	Version update to 10.0.1
4	12/2018	Version update for MCUXpresso SDK 2.5.0.
5	4/2019	Updated Section 6.1, "List of examples".

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile 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.

© 2019 NXP B.V.

Document Number MCUXSDKFRTOSGSUG
Revision 5, 04/2019