Application Report
SPNA237–April 2018

FreeRTOS on Hercules Devices

Veena Kamath

ABSTRACT

FREERTOS is a popular open-source real-time operating system used in embedded systems. It provides APIs for tasks, semaphores, mutexes, timers, and so forth.

HALCoGen is a GUI-based driver generating tool for the Hercules family of devices. HALCoGen also supports FREERTOS for various devices in the Hercules family. It enables users to generate the FREERTOS code, along with other drivers.

This application report provides an overview on FreeRTOS. This document is not a FreeRTOS user's guide, and only describes how to use FreeRTOS on Hercules devices with HALCoGen.

Contents

List of Figures

List of Tables

Trademarks

1 FreeRTOS Source Files

FreeRTOS source files can be divided into platform-dependent and platform-independent files. It supports various families, including TI's Hercules microcontroller.

The FreeRTOS\Source folder contains the drivers for each submodule, such as tasks, semaphores, queues, and so forth. These drivers are platform-independent.

The FreeRTOS\Source\include folder contains the header files, including a header file named mpu_wrappers.h. For the platforms that support MPU, these header files redefine the kernel functions to the corresponding MPU wrapper functions. More details on MPU wrapper functions are provided in the following sections.

The FreeRTOS\Source\portable\XXX folder contains platform-dependent code. For every platform, FREERTOS provides a .c and an assembly file for port functions, and a header file named portmacro.h for the platform-dependent macros.

Along with these source files, a FreeRTOS project requires the following files:

- FreeRTOSConfig.h — Contains macros for customizing the FreeRTOS drivers. Sample file provided as part of demos.
- mpu_wrappers.c — Contains the MPU wrapper functions for kernel functions. For FreeRTOS v9 and later, a sample file is provided as part of FreeRTOS\Source\portable\Common folder.

2 FreeRTOS Support in HALCoGen

To generate FreeRTOS code along with HALCoGen drivers, select DeviceName_FREERTOS as Device while creating the HALCoGen project, as shown in Figure 1. FreeRTOS drivers are not supported for every device variant in a device family. This document provides steps on how to customize the drivers for a different device variant.

The FreeRTOS source files are generated with a prefix "os_".

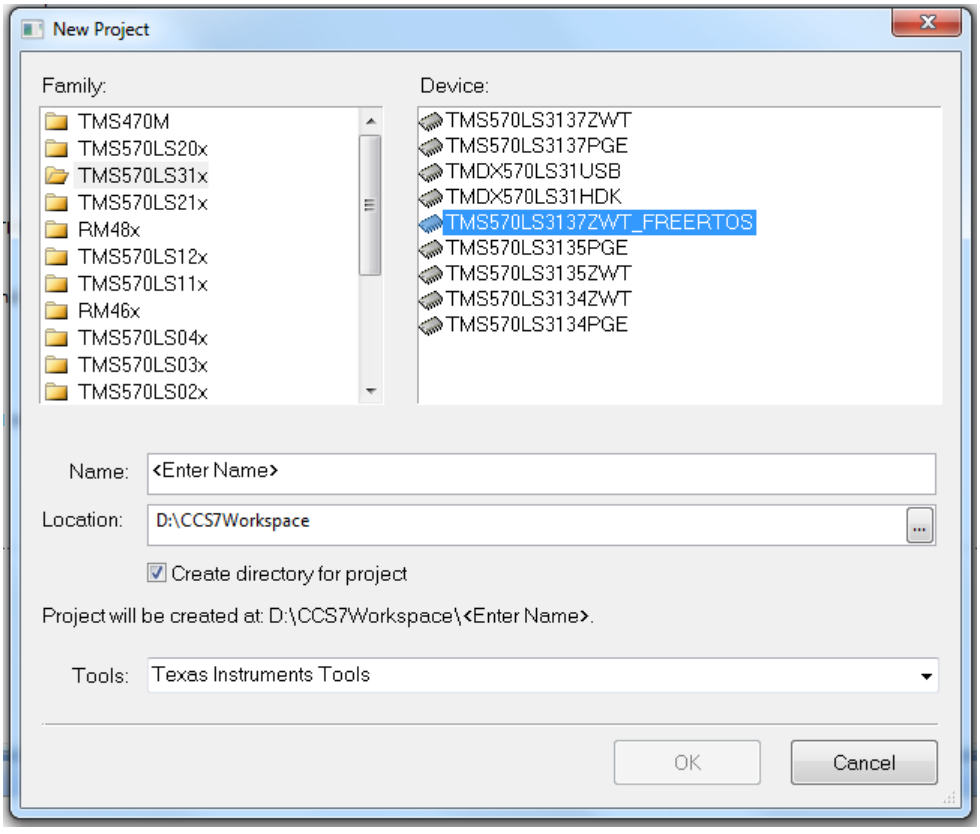

Figure 1. Select Device

Navigate to the OS tab to configure various options for the OS files, as shown in Figure 2.

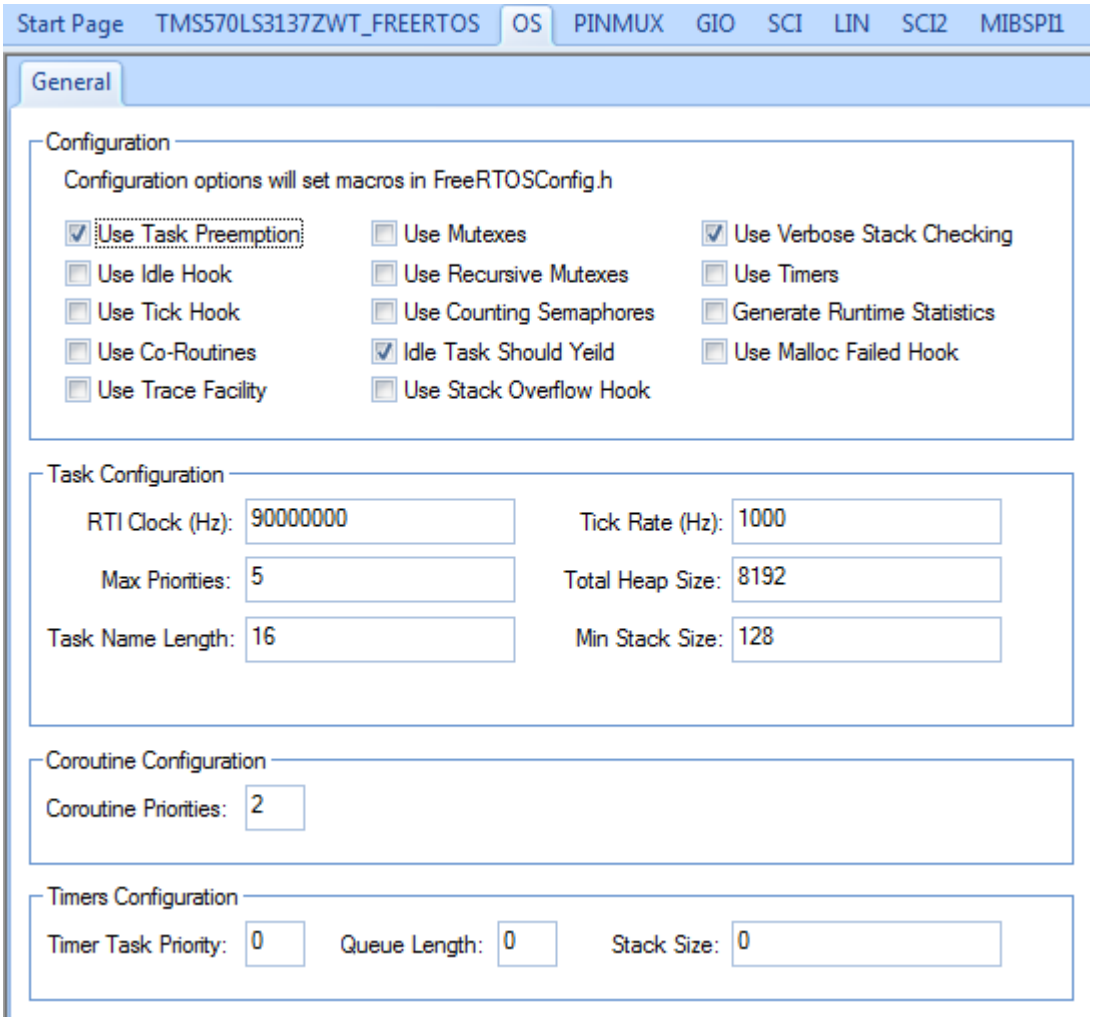

Figure 2. OS Tab

These configurations are translated to create the FreeRTOSConfig.h, and are provided as part of the generated drivers.

3 MPU Settings as an Add-on for Safety Use Cases

For devices with a Memory Protection Unit (MPU), FreeRTOS provides hooks for MPU configurations. MPU lets the user partition the device memory into various regions and configure the access rights for each regions. TI's Hercules microcontrollers are Arm® Cortex®-R based and support MPU.

3.1 MPU Wrapper Functions

There are two modes of CPU: privileged mode and user mode (names may differ depending on the device). User mode is a restricted mode, and is typically used to execute the user tasks. Privileged mode is used to execute the kernel code and for system initialization. The kernel code and data are typically placed in a privileged memory section, so that these are not accessed directly by user tasks. MPU wrapper functions provide wrappers for kernel functions so that they are executed in privileged mode.

Sample MPU wrapper function:

```
void MPU_vTaskDelete ( TaskHandle_t xTaskToDelete )
{
     BaseType_t xRunningPrivileged = prvRaisePrivilege();
     vTaskDelete ( xTaskToDelete );
     portRESET_PRIVILEGE( xRunningPrivileged );
}
```

3.2 MPU Regions for Tasks

The FreeRTOS task driver provides support to have n number of memory regions assigned to each task. The memory regions and its access rights may vary, depending on the tasks.

In a FreeRTOS environment, the MPU regions can be categorized into task-specific regions and common regions. Task-specific regions are configured every time a task switch occurs. The common regions are configured at the time the scheduler is initialized.

The first four regions and the last region (regions 0-3 and n-1) are configured when the scheduler is started and are common for all the tasks. This includes the flash, RAM, and peripherals memories. The other regions are task-specific, and can be configured during the task creation. There is a dedicated MPU region for the task stack.

Table 1 lists the device families with the number of MPU regions, and how they are split into common and task-specific regions.

Table 1. Device Families

Device Family	No. of MPU Regions	Common Regions	Task-Specific Regions
TMS570LS31x RM48x TMS570LS12x RM46x	12	Region 0: Unprivileged flash region Region 1: Privileged flash region Region 2: Privileged RAM region Region 3: Peripherals region Region 11: Privileged system region	Region 4: Task stack Regions 5-10: User-defined
TMS570LS04x RM42x	8	Region 0: Unprivileged flash region Region 1: Privileged flash region Region 2: Privileged RAM region Region 3: Peripherals region Region 7: Privileged system region	Region 4: Task stack Regions 5,6: User-defined
TMS570LC43x RM57x	16	Region 0: Unprivileged flash region Region 1: Privileged flash region Region 2: Privileged RAM region Region 3: Peripherals region Region 15: Privileged system region	The number of task-configurable regions can be changed in the HALCoGen GUI. Default = 3 Region 11: Task stack Regions 12-14: User-defined The remaining regions can be configured in the HALCoGen GUI, and are common for all tasks.

NOTE: Privileged flash and RAM regions include the privileged code and data section used by the kernel.

All the MPU configurations done through the HALCoGen GUI are reset when the scheduler is initialized. With TMS570LC43x and RM57x devices, there is an option to set the number of regions used by the FreeRTOS. The regions not used by FreeRTOS can be configured through the HALCoGen GUI. These are not reset by the FreeRTOS code.

To modify the default MPU regions set by FreeRTOS, modify the prvSetupDefaultMPU function available on the os_port.c file.

A task can be created using two FreeRTOS APIs:

- xTaskCreate — This disables all user-defined task-specific regions, and enables access to the entire RAM region as part of the task stack region. The task created using this API is called a non-restricted task, as this has access to the entire RAM region.
- xTaskCreateRestricted — This lets the user define the regions they need access to. The task stack region is set with the actual stack size. The task created using this API is called a restricted task, as this does not have access to any RAM region other than its own stack (unless explicitly mentioned).

For more details on how to create restricted and non-restricted tasks, see the HALCoGen examples.

4 Porting FREERTOS on a Different Part Number

HALCoGen supports FreeRTOS driver generation for only a few devices. This section explains how to generate FreeRTOS drivers for any device part number. This example uses the TMS570LS1227PGE device. For the TMS570LS12x family, HALCoGen provides a FreeRTOS port for the TMS570LS1227ZWT part.

1. Create a HALCoGen project with TMS570LS1227ZWT_FREERTOS, as shown in Figure 3.

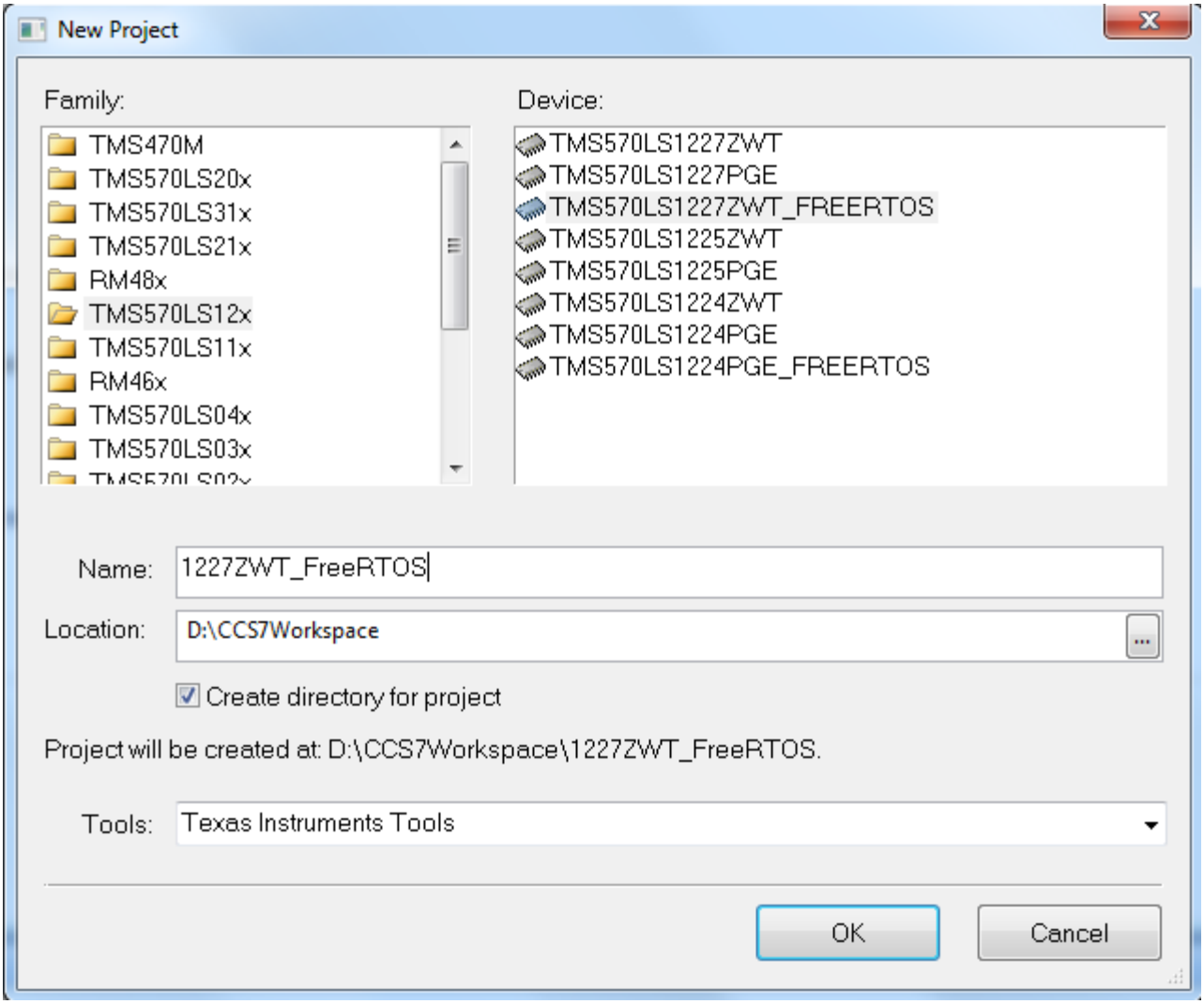

Figure 3. Create a HALCoGen Project

2. Configure the PLL and other clock settings as per the actual device requirements. In the example shown in Figure 4, the maximum CPU clock for the TMS570LS1227PGE is 160 MHz, whereas, with the TMS570LS1227ZWT, it is 180 MHz.

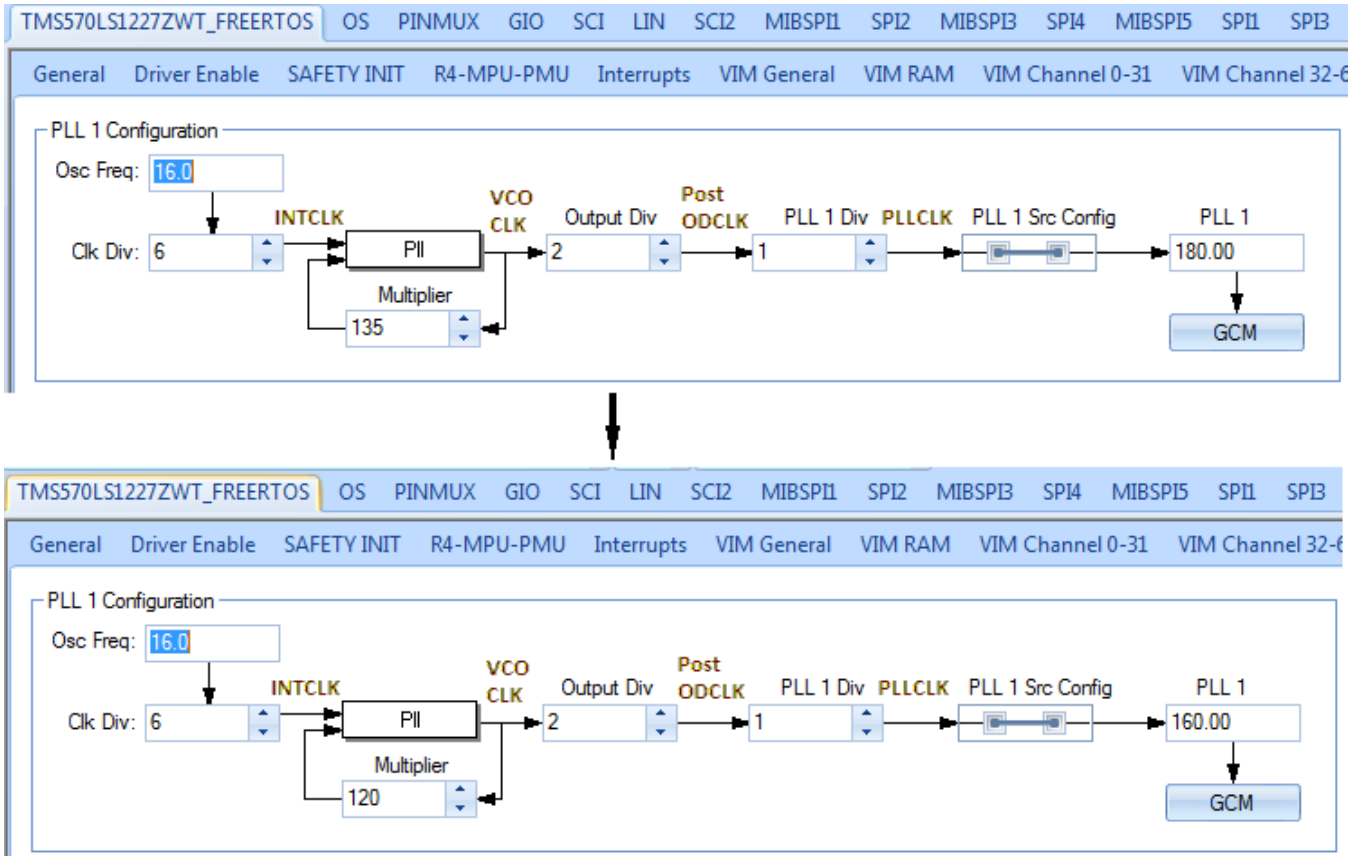

Figure 4. Configure PLL

3. Configure the FreeRTOS port as required in the OS tab. Save and generate code.

4. Create a HALCoGen project with the TMS570LS1227PGE, as shown in Figure 5.

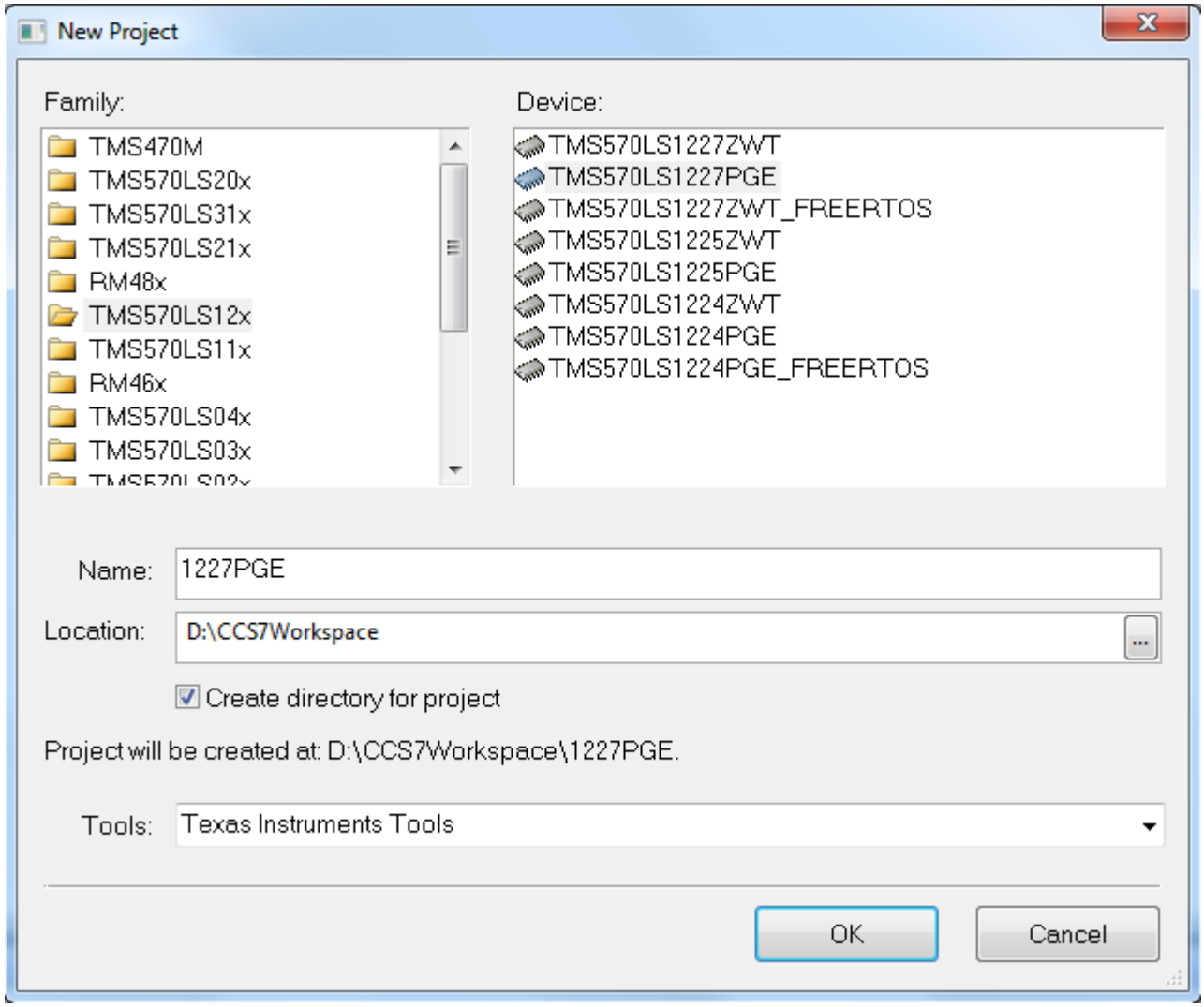

Figure 5. New Project

5. Set VIM configurations as shown in Figure 6. FreeRTOS uses 2 interrupts: RTI Compare 0 and System Software Interrput (SSI) (see Figure 7).

   Enable interrupt channel: 2 - RTI Compare 0. Set ISR name as vPortPreemptiveTick

   Enable interrupt channel: 21 - SSI. Set ISR name as vPortYeildWithinAPI.

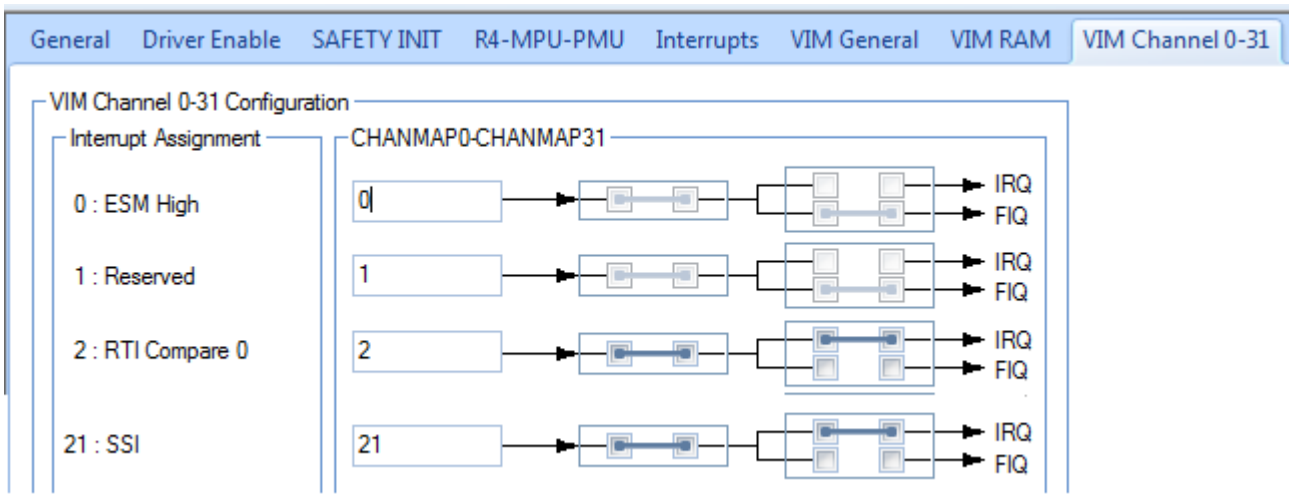

Figure 6. VIM Configuration

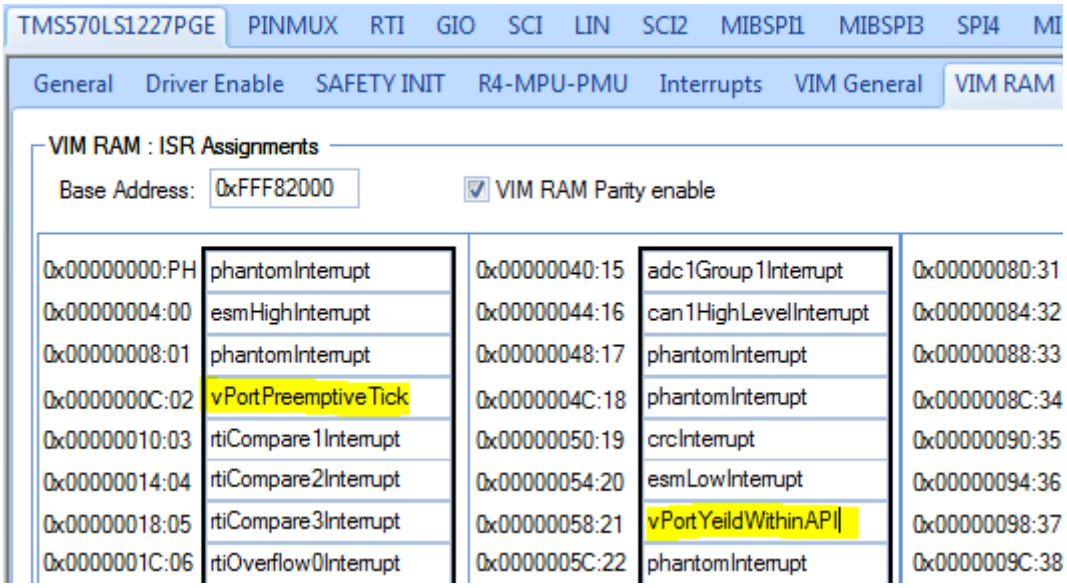

Figure 7. VIM ISR Assignments

6. FreeRTOS also uses the SVC exception. Enable the exception handler as vPortSWI, as shown in Figure 8.

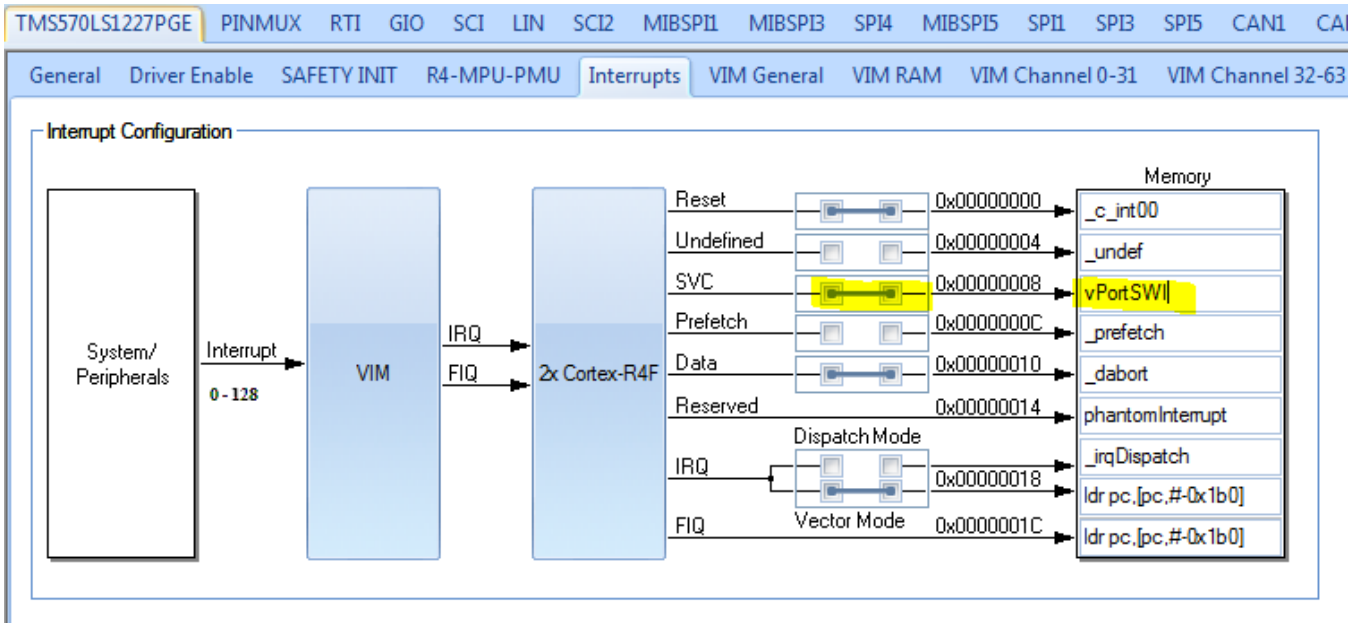

Figure 8. Enable PortSWI

7. Disable the RTI driver and configure the other peripherals as required, then generate the code.
8. Copy the OS files from the 1227ZWT project to the 1227PGE project. This includes the files starting with prefix "os_", FreeRTOS.h and FreeeRTOSConfig.h.
9. Make the following changes in the linker command file in 1227PGE project:

```
MEMORY
{
    VECTORS (X)  : origin=0x00000000 length=0x00000020
    KERNEL  (RX) : origin=0x00000020 length=0x00008000
    FLASH0  (RX) : origin=0x00008020 length=0x0013FFE0 - 0x8000
    STACKS  (RW) : origin=0x08000000 length=0x00001500
    KRAM    (RW) : origin=0x08001500 length=0x00000800
    RAM     (RW) : origin=0x08001500 + 0x800 length=0x0002EB00 - 0x800

/* USER CODE BEGIN (2) */
/* USER CODE END */
}

/* USER CODE BEGIN (3) */
/* USER CODE END */

/*----------------------------------------------------------------------------
/* Section Configuration

SECTIONS
{
    .intvecs : {} > VECTORS
    /* FreeRTOS Kernel in protected region of Flash */
    .kernelTEXT   : {} > KERNEL
    .cinit        : {} > KERNEL
    .pinit        : {} > KERNEL
    /* Rest of code to user mode flash region */
    .text         : {} > FLASH0
    .const        : {} > FLASH0
    /* FreeRTOS Kernel data in protected region of RAM */
    .kernelBSS    : {} > KRAM
    .kernelHEAP   : {} > RAM
    .bss          : {} > RAM
    .data         : {} > RAM

/* USER CODE BEGIN (4) */
/* USER CODE END */
}
```

10. The FreeRTOS port is now ready for the TMS570LS1227PGE device.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265